Delft University of Technology

A max-cut approximation using a graph based MBO scheme

Keetch, Blaine; van Gennip, Yves

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

A MAX-CUT APPROXIMATION
USING A GRAPH BASED MBO SCHEME

Blaine Keetch and Yves van Gennip

School of Mathematical Sciences
The University of Nottingham
University Park, Nottingham, United Kingdom, NG7 2RD

(Communicated by Lee DeVille))

Abstract. The Max-Cut problem is a well known combinatorial optimization
problem. In this paper we describe a fast approximation method. Given a
graph G, we want to find a cut whose size is maximal among all possible cuts.
A cut is a partition of the vertex set of G into two disjoint subsets. For an
unweighted graph, the size of the cut is the number of edges that have one
vertex on either side of the partition; we also consider a weighted version of
the problem where each edge contributes a nonnegative weight to the cut.

We introduce the signless Ginzburg–Landau functional and prove that this
functional Γ-converges to a Max-Cut objective functional. We approximately
minimize this functional using a graph based signless Merriman–Bence–Osher
(MBO) scheme, which uses a signless Laplacian. We derive a Lyapunov func-
tional for the iterations of our signless MBO scheme. We show experimentally
that on some classes of graphs the resulting algorithm produces more accurate
maximum cut approximations than the current state-of-the-art approximation
algorithm. One of our methods of minimizing the functional results in an al-
gorithm with a time complexity of $\mathcal{O}(|E|)$, where $|E|$ is the total number of
edges on G.

1. **Introduction.**

1.1. **Maximum cut.** Given an undirected (edge-)weighted graph $G = (V, E, \omega)$, a
cut $V_{-1}|V_1$ is a partition of the node set V into two disjoint subsets V_{-1} and V_1.
The size of a cut $C = V_{-1}|V_1$, denoted by $s(C)$, is the sum of all the weights corre-
sponding to edges that have one end vertex in V_{-1} and one in V_1. The maximum
cut (Max-Cut) problem is the problem of finding a cut C^* such that for all cuts
C, $s(C) \leq s(C^*)$. We call such a C^* a maximum cut and say $\mathrm{mc}(G) := s(C^*)$ is
the maximum cut value of the graph G. The Max-Cut problem for an unweighted
graph is a special case of the Max-Cut problem on a weighted graph which we ob-
tain by assuming all edge weights are 1. Finding an unweighted graph's Max-Cut is
equivalent to finding a bipartite subgraph with the largest number of edges possible.
In fact, for an unweighted bipartite graph $\mathrm{mc}(G) = |E|$.

The Max-Cut problem is an NP-hard problem; assuming P $\neq$ NP no solution can be acquired in polynomial time. There are a variety of polynomial time approximation algorithms for this problem [25, 15, 53]. Some Max-Cut approximation algorithms have a proven lower bound on their accuracy, which asserts the existence of a $\beta \in [0,1]$ such that, for all output cuts C obtained by the algorithm, $s(C) \geq \beta \mathrm{mc}(G)$. We call such a β a performance guarantee. For algorithms that incorporate stochastic steps, such a lower bound typically takes the form $E[s(C)] \geq \beta \mathrm{mc}(G)$ instead, where $E[s(C)]$ denotes the expected value of the size of the output cut.

In recent years a new type of approach to approximating such graph problems has gained traction. Models from the world of partial differential equations and variational methods that exhibit behaviour of the kind that could be helpful in solving the graph problem are transcribed from their usual continuum formulation to a graph based model. The resulting discrete model can then be solved using techniques from numerical analysis and scientific computing. Examples of problems that have successfully been tackled in this manner include data classification [11], image segmentation [16], and community detection [33]. In this paper we use a variation on the graph Ginzburg–Landau functional, which was introduced in [11], to construct an algorithm which approximately solves the Max-Cut problem on simple undirected weighted graphs.

We compare our method with the Goemans–Williamson (GW) algorithm [25], which is the current state-of-the-art method for approximately solving the Max-Cut problem. In [25] the authors solve a relaxed Max-Cut objective function and intersect the solution with a random hyperplane in a n-dimensional sphere. It is proven that if $\mathrm{gw}(C)$ is the size of the cut produced by the Goemans–Williamson algorithm, then its expected value $E[\mathrm{gw}(C)]$ satisfies the inequality $E[\mathrm{gw}(C)] \geq \beta \mathrm{mc}(G)$ where $\beta = 0.878$ (rounded down). If the Unique Games Conjecture [36] is true, the GW algorithm has the best performance guarantee that is possible for a polynomial time approximation algorithm [37]. It has been proven that approximately solving the Max-Cut problem with a performance guarantee of $\frac{16}{17} \approx 0.941$ or better is NP-hard [52].

Finding $\mathrm{mc}(G)$ is equivalent to finding the ground state of the Ising Hamiltonian in Ising spin models [31], and 0/1 linear programming problems can be restated as Max-Cut problems [38].

1.2. **Signless Ginzburg–Landau functional.** Spectral graph theory [18] explores the relationships between the spectra of graph operators, such as graph Laplacians (see Section 2.1), and properties of graphs. For example, the multiplicity of the zero eigenvalue of the (unnormalised, random walk, or symmetrically normalised) graph Laplacian is equal to the number of connected components of the graph. Such properties lie at the basis of the successful usage of the graph Laplacian in graph clustering, such as in spectral clustering [58] and in clustering and classification methods that use the graph Ginzburg–Landau functional [11]

$$f_\varepsilon(u) := \frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i - u_j)^2 + \frac{1}{\varepsilon} \sum_{i \in V} W(u_i). \qquad (1)$$

Here $u : V \to \mathbb{R}$ is a real-valued function defined on the node set V, with value u_i on node i, ω_{ij} is a positive weight associated with the edge between nodes i and j (and $\omega_{ij} = 0$ if such an edge is absent), and $W(x) := (x^2 - 1)^2$ is a double-well

potential with minima at $x = \pm 1$. In Section 2.1 we will introduce our setting and notation more precisely.

The method we use in this paper is based on a variation of f_ε, we call the *signless Ginzburg–Landau functional*:

$$f_\varepsilon^+(u) := \frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i + u_j)^2 + \frac{1}{\varepsilon} \sum_{i \in V} W(u_i). \tag{2}$$

This nomenclature is suggested by the fact that the signless graph Laplacians are related to f_ε^+ in a similar way as the graph Laplacians are related to f_ε, as we will see in Section 2.1. Signless graph Laplacians have been studied because of the connections between their spectra and bipartite subgraphs [21]. In [32, 55] the authors derive a graph difference operator and a graph divergence operator to form a graph Laplacian operator. In this paper we mimic this framework by deriving a signless difference operator and a signless divergence operator to form a signless Laplacian operator. Whereas the graph Laplacian operator is a discretization of the continuum Laplacian operator, the continuum analogue of the signless Laplacian is the subject of current and future research. Preliminary calculations for specific graph structures suggest that the continuum signless Laplacian is an integral operator [34].

The functional f_ε is useful in clustering and classification problems, because minimizers of f_ε (in the presence of some constraint or additional term, to prevent trivial minimizers) will be approximately binary (with values close to ± 1), because of the double-well potential term, and will have similar values on nodes that are connected by highly weighted edges, because of the first term in f_ε. This intuition can be formalised using the language of Γ-convergence [20]. In analogy with the continuum case in [44, 45], it was proven in [55] that if $\varepsilon \downarrow 0$, then f_ε Γ-converges to

$$f_0(u) := \begin{cases} 2\mathrm{TV}(u), & \text{if } u \text{ only takes the values } \pm 1, \\ \infty, & \text{otherwise}, \end{cases}$$

where $\mathrm{TV}(u) := \frac{1}{2} \sum_{i,j \in V} \omega_{ij} |u_i - u_j|$ is the graph total variation[1]. Together with an equi-coercivity property, which we will return to in more detail in Section 4, this Γ-convergence result guarantees that minimizers of f_ε converge to minimizers of f_0 as $\varepsilon \downarrow 0$. If u only takes the values ± 1, we note that $\mathrm{TV}(u) = 2s(C)$, where $C = V_{-1}|V_1$ is the cut given by $V_{\pm 1} := \{i \in V : u_i = \pm 1\}$. Hence minimizers u^ε of f_ε are expected to approximately solve the *minimal* cut problem, if we let $V_{\pm 1} \approx \{i \in V : u_i^\varepsilon \approx \pm 1\}$.

In Section 4 we prove that f_ε^+ Γ-converges to a limit functional whose minimizers solve the Max-Cut problem. Hence, we expect minimizers u^ε of f_ε^+ to approximately solve the Max-Cut problem, if we consider the cut $C = V_{-1}|V_1$, with $V_{\pm 1} = \{i \in V : u_i^\varepsilon \approx \pm 1\}$.

1.3. **Graph MBO scheme.** There are various ways in which the minimization of f_ε^+ can be attempted. One such way, which can be explored in a future publication, is to use a gradient flow method. In the case of f_ε the gradient flow is given by an Allen–Cahn type equation on graphs [11, 56],

---

[1]The multiplicative factor 2 in f_0 above differs from that in [55] because in the current paper we choose different locations for the wells of W.

$$\frac{du_i}{dt} = -(\Delta u)_i - \frac{1}{\varepsilon} d_i^{-r} W'(u_i), \tag{3}$$

where Δu is a graph Laplacian of u, d_i the degree of node i, and $r \in [0, 1]$ a parameter (see Section 2.1 for further details). This can be solved using a combination of convex splitting and spectral truncation. In the case of f_ε^+ such an approach would lead to a similar equation and scheme, with the main difference being the use of a signless graph Laplacian instead of a graph Laplacian.

In this paper, however, we have opted for an alternative approach, which is also inspired by similar approaches which have been developed for the f_ε case. The continuum Merriman–Bence–Osher (MBO) scheme [41, 42] involves iteratively solving the diffusion equation over a small time step τ and thresholding the solution to an indicator function. For a short diffusion time τ this scheme approximates motion by mean curvature [10]. This scheme has been adapted to a graph setting [40, 56]. Heuristically it is expected that the outcome of the graph MBO scheme closely approximates minimizers of f_ε, as the diffusion step involves solving $\frac{du_i}{dt} = -(\Delta u)_i$ and the thresholding step has a similar effect as the nonlinearity $-\frac{1}{\varepsilon} W'(u_i)$ in (3). Experimental results strengthen this expectation, however rigorous confirmation is still lacking.

In order to approximately minimize f_ε^+, and consequently approximately solve the Max-Cut problem, we use an MBO type scheme in which we replace the graph Laplacian in the diffusion step by a signless graph Laplacian. We use two methods to compute this step: (1) a spectral method, adapted from the one in [11], which allows us to use a small subset of the eigenfunctions, which correspond to the smallest eigenvalues of the graph Laplacian, and (2) an Euler method. Moreover, we show (Section 6) that this variant of the graph MBO scheme has a Lyapunov functional associated with it which Γ-converges to a Max-Cut objective functional (i.e. a functional whose minimizers solve the Max-Cut problem) as the signless 'diffusion' time $\tau \downarrow 0$.

The usefulness of (normalised) signless graph Laplacians when attempting to find maximum cuts can be intuitively understood from the fact that their spectra are in a sense (which is made precise in Proposition 2) the reverse of the spectra of the corresponding (normalised) graph Laplacians. Hence, where a standard graph Laplacian driven diffusion leads to clustering patterns according to the eigenfunctions corresponding to its smallest eigenvalues, 'diffusion' driven by a signless graph Laplacian leads to patterns resembling the eigenfunctions corresponding to the smallest eigenvalues of that signless graph Laplacian and thus the largest eigenvalues of the corresponding standard graph Laplacian.

1.4. **Structure of the paper.** In Section 2 we explain the notation we use in this paper and give some preliminary results. Section 3 gives a precise formulation of the Max-Cut problem and discusses the Goemans–Williamson algorithm in more detail. In Section 4 we introduce the signless graph Ginzburg–Landau functional f_ε^+ and use Γ-convergence techniques to prove that minimizers of f_ε^+ can be used to find approximate maximum cuts. We describe the signless MBO algorithm we use to find approximate minimizers of f_ε^+ in Section 5 and discuss the results we get in Section 7. In Section 6 we derive a Lyapunov functional whose value is non-increasing along (MBO+) iterations, and show that subject to a re-scaling this functional Γ-converges to a total variation objective functional, which is shown to be related to maximum cuts. We analyse the influence of our parameter choices in Section 8 and conclude the paper in Section 9.

2. Setup and notation.

2.1. **Graph based operators and functionals.** In this paper we will consider non-empty finite, simple[2], undirected graphs $G = (V, E, \omega)$ without isolated nodes, with vertex set (or node set) V, edge set $E \subset V^2$ and non-negative edge weights ω. We denote the set of all such graphs by $\mathcal{G}$. By assumption V has finite cardinality, which we denote by $n := |V| \in \mathbb{N}$[3]. We assume a node labelling such that $V = \{1, \ldots, n\}$. When $i, j \in V$ are nodes, the undirected edge between i and j, if present, is denoted by (i, j). The edge weight corresponding to this edge is $\omega_{ij} > 0$. Since G is undirected, we identify (i, j) with (j, i) in E. Within this framework we can also consider unweighted graphs, which correspond to the cases in which, for all $(i, j) \in E$, $\omega_{ij} = 1$.

We define $\mathcal{V}$ to be the set consisting of all node functions $u : V \to \mathbb{R}$ and $\mathcal{E}$ to be the set of edge functions $\varphi : E \to \mathbb{R}$. We will use the notation $u_i := u(i)$ and $\varphi_{ij} := \varphi(i, j)$ for functions $u \in \mathcal{V}$ and $\varphi \in \mathcal{E}$, respectively. For notational convenience, we will typically associate $\varphi \in \mathcal{E}$ with its extension to V^2 obtained by setting $\varphi_{ij} = 0$ if $(i, j) \notin E$. We also extend ω to V^2 in this way: if $(i, j) \notin E$, then $\omega_{ij} = 0$. Because $G \in \mathcal{G}$ is undirected, we have for all $(i, j) \in E$, $\omega_{ij} = \omega_{ji}$. Because $G \in \mathcal{G}$ is simple, for all $i \in V$, $(i, i) \notin E$. The degree of a node i is $d_i := \sum_{j \in V} \omega_{ij}$. Because $G \in \mathcal{G}$ does not contain isolated nodes, we have for all $i \in V, d_i > 0$.

As shown in [32], it is possible for $\mathcal{V}$ and $\mathcal{E}$ to be defined for directed graphs, but we will not pursue these ideas here.

To introduce the graph Laplacians and signless graph Laplacians we use and extend the structure that was used in [32, 55, 56]. We define the inner products on $\mathcal{V}$ and $\mathcal{E}$ as

$$\langle u, v \rangle_{\mathcal{V}} := \sum_{i \in V} u_i v_i d_i^r, \qquad \langle \varphi, \phi \rangle_{\mathcal{E}} := \frac{1}{2} \sum_{i, j \in V} \varphi_{ij} \phi_{ij} \omega_{ij}^{2q-1},$$

where $r \in [0, 1]$ and $q \in [\frac{1}{2}, 1]$. If $r = 0$ and $d_i = 0$, we interpret d_i^r as 0. Similarly for ω_{ij}^{2q-1} and other such expressions below.

We define the graph gradient operator $(\nabla : \mathcal{V} \to \mathcal{E})$ by, for all $(i, j) \in E$,

$$(\nabla u)_{ij} := \omega_{ij}^{1-q}(u_j - u_i).$$

We define the graph divergence operator $(\mathrm{div} : \mathcal{E} \to \mathcal{V})$ as the adjoint of the gradient, and a graph Laplacian operator $(\Delta_r : \mathcal{V} \to \mathcal{V})$ as the graph divergence of the graph gradient: for all $i \in V$,

$$(\mathrm{div}\varphi)_i := \frac{1}{2} d_i^{-r} \sum_{j \in V} \omega_{ij}^q (\varphi_{ji} - \varphi_{ij}), \qquad (\Delta_r u)_i := (\mathrm{div}(\nabla u))_i = d_i^{-r} \sum_{j \in V} \omega_{ij}(u_i - u_j).$$

(4)

We note that the choices $r = 0$ and $r = 1$ lead to Δ_r being the unnormalised graph Laplacian and random walk graph Laplacian, respectively [46, 58]. Hence it is useful for us to explicitly incorporate r in the notation Δ_r for the graph Laplacian.

In analogy with the graph gradient, divergence, and Laplacian, we now define their 'signless' counterparts. We define the signless gradient operator $(\nabla^+ : \mathcal{V} \to \mathcal{E})$ by, for all $(i, j) \in E$,

$$(\nabla^+ u)_{ij} := \omega_{ij}^{1-q}(u_j + u_i).$$

---

[2]By 'simple' we mean 'without self-loops and without multiple edges between the same pair of vertices'. Note that removing self-loops from a graph does not change its maximum cut.

[3]For definiteness we use the convention $0 \notin \mathbb{N}$.

Then we define the signless divergence operator $(\text{div}^+ : \mathcal{E} \to \mathcal{V})$ to be the adjoint of the signless gradient, and the signless Laplacian operator $(\Delta_r^+ : \mathcal{V} \to \mathcal{V})$ as the signless divergence of the signless gradient[4]: for all $i \in V$,

$$(\text{div}^+\varphi)_i := \frac{1}{2}d_i^{-r}\sum_{j\in V}\omega_{ij}^q(\varphi_{ji}+\varphi_{ij}), \ (\Delta_r^+ u)_i := (\text{div}^+(\nabla^+ u))_i = d_i^{-r}\sum_{j\in V}\omega_{ij}(u_i+u_j).$$

By definition we have

$$\langle \nabla u, \phi \rangle_\mathcal{E} = \langle u, \text{div}\,\phi \rangle_\mathcal{V}, \qquad \langle \nabla^+ u, \phi \rangle_\mathcal{E} = \langle u, \text{div}^+\phi \rangle_\mathcal{V}.$$

Proposition 1. *The operators* $\Delta_r : \mathcal{V} \to \mathcal{V}$ *and* $\Delta_r^+ : \mathcal{V} \to \mathcal{V}$ *are self-adjoint and positive-semidefinite.*

Proof. Let $u, v \in \mathcal{V}$. Since $\langle u, \Delta_r v \rangle_\mathcal{V} = \langle \nabla u, \nabla v \rangle_\mathcal{E} = \langle \Delta_r u, v \rangle_\mathcal{V}$ and $\langle u, \Delta_r^+ v \rangle_\mathcal{V} = \langle \nabla^+ u, \nabla^+ v \rangle_\mathcal{E} = \langle \Delta_r^+ u, v \rangle_\mathcal{V}$, the operators are self-adjoint. Positive-semidefiniteness follows from $\langle u, \Delta_r u \rangle_\mathcal{V} = \langle \nabla u, \nabla u \rangle_\mathcal{E} \geq 0$ and $\langle u, \Delta_r^+ u \rangle_\mathcal{V} = \langle \nabla^+ u, \nabla^+ u \rangle_\mathcal{E} \geq 0$. $\qquad\square$

In the literature a third graph Laplacian is often used, besides the unnormalised and random walk graph Laplacians. This symmetrically normalised graph Laplacian [18] is defined by, for all $i \in V$,

$$(\Delta_s u)_i := \frac{1}{\sqrt{d_i}}\sum_{j\in V}\omega_{ij}\left(\frac{u_i}{\sqrt{d_i}} - \frac{u_j}{\sqrt{d_j}}\right).$$

This Laplacian cannot be obtained by choosing a suitable r in the framework we introduced above, but will be useful to consider in practical applications. Analogously, we define the signless symmetrically normalised graph Laplacian by, for all $i \in V$,

$$(\Delta_s^+ u)_i := \frac{1}{\sqrt{d_i}}\sum_{j\in V}\omega_{ij}\left(\frac{u_i}{\sqrt{d_i}} + \frac{u_j}{\sqrt{d_j}}\right).$$

There is a canonical way to represent a function $u \in \mathcal{V}$ by a vector in $\mathbb{R}^n$ with components u_i. The operators Δ_r and Δ_r^+ can then be represented by the $n \times n$ matrices $L_r := D^{1-r} - D^{-r}A$ and $L_r^+ := D^{1-r} + D^{-r}A$, respectively. Here D is the degree matrix, i.e. the diagonal matrix with diagonal entries $D_{ii} := d_i$, and A is the weighted adjacency matrix with entries $A_{ij} := \omega_{ij}$.

Similarly the operators Δ_s and Δ_s^+ are then represented by $L_s := I - D^{-1/2}AD^{-1/2}$ and $L_s^+ := I + D^{-1/2}AD^{-1/2}$, respectively, where I denotes the $n \times n$ identity matrix. Any eigenvalue-eigenvector pair (λ, v) of L_r, L_r^+, L_s, L_s^+ corresponds via the canonical representation to an eigenvalue-eigenfunction pair (λ, ϕ) of Δ_r, Δ_r^+, Δ_s, Δ_s^+, respectively. We refer to the eigenvalue-eigenvector pair (λ, v) as an eigenpair.

For a vertex set $S \subset V$, we define the indicator function (or characteristic function)

$$\chi_S := \begin{cases} 1, & \text{if } i \in S, \\ 0, & \text{if } i \notin S. \end{cases}$$

---

[4]In some papers the space $\mathcal{E}$ is defined as the space of all *skew-symmetric* edge functions. We do not require the skew-symmetry condition here, hence $\nabla^+ u \in \mathcal{E}$, having div^+ act on $\nabla^+ u$ is consistent with our definitions, and $\text{div}^+\varphi$ is not identically equal to 0 for all $\varphi \in \mathcal{E}$.

We define the inner product norms $\|u\|_{\mathcal{V}} := \sqrt{\langle u, u \rangle_{\mathcal{V}}}$, $\|\phi\|_{\mathcal{E}} := \sqrt{\langle \phi, \phi \rangle_{\mathcal{E}}}$ which we use to define the Dirichlet energy and signless Dirichlet energy,

$$\frac{1}{2}\|\nabla u\|_{\mathcal{E}}^2 = \frac{1}{4}\sum_{i,j \in V} \omega_{ij}(u_i - u_j)^2 \quad \text{and} \quad \frac{1}{2}\|\nabla^+ u\|_{\mathcal{E}}^2 = \frac{1}{4}\sum_{i,j \in V} \omega_{ij}(u_i + u_j)^2.$$

In particular we recognise that the graph Ginzburg–Landau functional $f_\varepsilon : \mathcal{V} \to \mathbb{R}$ from (1) and the signless graph Ginzburg–Landau functional $f_\varepsilon^+ : \mathcal{V} \to \mathbb{R}$ from (2) can be written as

$$f_\varepsilon(u) = \|\nabla u\|_{\mathcal{E}}^2 + \frac{1}{\varepsilon}\sum_{i \in V} W(u_i) \quad \text{and} \quad f_\varepsilon^+(u) = \|\nabla^+ u\|_{\mathcal{E}}^2 + \frac{1}{\varepsilon}\sum_{i \in V} W(u_i).$$

It is interesting to note here an important difference between the functionals f_ε and f_ε^+. Most of the results that are derived in the literature for f_ε (such as the Γ-convergence results in [55]) do not crucially depend on the specific locations of the wells of W. For example, in f_ε the wells are often chosen to be at 0 and 1, instead of at -1 and 1. However, for f_ε^+ we have less freedom to choose the wells without drastically altering the properties of the functional. The wells have to be placed symmetrically with respect to 0, because we want $(u_i + u_j)^2$ to be zero when u_i and u_j are located in different wells. In particular, we see that placing a well at 0 would have the undesired consequence of introducing the trivial minimizer $u = 0$. This points to a second, related, difference. Whereas minimization of f_ε in the absence of any further constraints or additional terms in the functional leads to trivial minimizers of the form $u = c\chi_V$, where $c \in \mathbb{R}$ is one of the values of the wells of W (so $c \in \{-1, 1\}$ for our choice of W), minimizers of f_ε^+ are not constant, if the graph has more than one vertex. The following lemma gives the details.

Lemma 2.1. *Let $G \in \mathcal{G}$ with $n \geq 2$, let $\varepsilon > 0$, and let u be a minimizer of $f_\varepsilon^+ : \mathcal{V} \to \mathbb{R}$ as in (2). Then u is not a constant function.*

Proof. Let $c \in \mathbb{R}$ and $i^* \in V$. Define the functions $u, \bar{u} \in \mathcal{V}$ by $u := c\chi_V$ and

$$\bar{u}_i := \begin{cases} c, & \text{if } i \neq i^*, \\ -c, & \text{if } i = i^*. \end{cases}$$

Since W is an even function, we have $\sum_{i \in V} W(\bar{u}_i) = \sum_{i \in V} W(u_i)$. Moreover, since for all $j \in V$, $\omega_{i^* j} = 0$ or $u_j = -u_{i^*}$, we have

$$\|\nabla^+ \bar{u}\|_{\mathcal{E}}^2 = \frac{1}{2}\sum_{\substack{i \in V \\ i \neq i^*}} \sum_{j \in V} \omega_{ij}(2c)^2 < \frac{1}{2}\sum_{i,j \in V} \omega_{ij}(2c)^2 = \|\nabla^+ u\|_{\mathcal{E}}^2.$$

The inequality is strict, because per assumption G has no isolated nodes and thus there is a $j \in V$ such that $\omega_{i^* j} > 0$. We conclude that $f_\varepsilon^+(\bar{u}) < f_\varepsilon^+(u)$, which proves that u is not a minimizer of f_ε^+. $\qquad\square$

We define the graph total variation $\text{TV} : \mathcal{V} \to \mathbb{R}$ as

$$\text{TV}(u) := \max\{\langle u, \text{div}\, \varphi \rangle_{\mathcal{V}} : \varphi \in \mathcal{E}, \forall i, j \in V \; |\varphi_{ij}| \leq 1\} = \frac{1}{2}\sum_{i,j \in V} \omega_{ij}^q |u_i - u_j|. \quad (5)$$

The second expression follows since the maximum in the definition is achieved by $\varphi = \text{sgn}(\nabla u)$ [56]. We can define an analogous (signless total variation) functional $\text{TV}^+ : \mathcal{V} \to \mathbb{R}$, using the signless divergence:

$$\text{TV}^+(u) := \max\{\langle u, \text{div}^+\, \varphi \rangle_{\mathcal{V}} : \varphi \in \mathcal{E}, \forall i, j \in V \; |\varphi_{ij}| \leq 1\}.$$

Lemma 2.2. *Let $u \in \mathcal{V}$, then $\mathrm{TV}^+(u) = \frac{1}{2}\sum_{i,j\in V}\omega_{ij}^q|u_i + u_j|$.*

Proof. Let $\varphi \in \mathcal{E}$ such that, for all $i, j \in V$, $|\varphi_{ij}| \leq 1$. We compute

$$\langle u, \mathrm{div}^+ \varphi\rangle_{\mathcal{V}} = \frac{1}{2}\sum_{i,j\in V}\omega_{ij}^q u_i(\varphi_{ji} + \varphi_{ij}) = \frac{1}{2}\sum_{i,j\in V}\omega_{ij}^q\varphi_{ij}(u_i + u_j)$$

$$\leq \frac{1}{2}\sum_{i,j\in V}\omega_{ij}^q|\varphi_{ij}||u_i + u_j| \leq \frac{1}{2}\sum_{i,j\in V}\omega_{ij}^q|u_i + u_j|.$$

Moreover, since $\varphi = \mathrm{sgn}(\nabla^+ u)$ is an admissable choice for φ and

$$\langle \mathrm{sgn}\left(\nabla^+ u\right), \mathrm{div}^+ \varphi\rangle_{\mathcal{V}} = \frac{1}{2}\sum_{i,j\in V}\omega_{ij}^q|u_i + u_j|,$$

the result follows. $\square$

Note that the total variation functional that was mentioned in Section 1.2 corresponds to the choice $q = 1$ in (5). This is the relevant choice for this paper and hence from now on we will assume that $q = 1$. Note that the choice of q does not have any influence on the form of the graph (signless) Laplacians.

One consequence of the choice $q = 1$ is that TV and TV^+ are now closely connected to cut sizes: If $S \subset V$ and $C = S|S^c$ is the cut induced by S, then

$$\mathrm{TV}\left(\chi_S - \chi_{S^c}\right) = 2\mathrm{TV}\left(\chi_S\right) = 2s(C) \quad \text{and} \quad \mathrm{TV}^+\left(\chi_S - \chi_{S^c}\right) = \sum_{i,j\in V}\omega_{ij} - 2s(C).$$

(6)

We will give a precise definition of $s(C)$ in Definition 3.1 below.

Definition 2.3. Let $G \in \mathcal{G}$. Then G is bipartite if and only if there exist $A \subset V$, $B \subset V$, such that all the conditions below are satisfied:
- $A \cup B = V$,
- $A \cap B = \emptyset$, and
- for all $(i, j) \in E$, $i \in A$ and $j \in B$, or $i \in B$ and $j \in A$.

In that case we say that G has a bipartition (A, B).

Definition 2.4. An Erdös-Rényi graph $G(n, p)$ is a realization of a random graph generated by the Erdös-Rényi model, i.e. it is an unweighted, undirected, simple graph with n nodes, in which, for all unordered pairs $\{i, j\}$ of distinct $i, j \in V$, an edge $(i, j) \in E$ has been generated with probability $p \in [0, 1]$.

2.2. **Spectral properties of the (signless) graph Laplacians.** We consider the Rayleigh quotients for Δ_r and Δ_r^+ defined, for $u \in V$, as

$$R(u) := \frac{\langle u, \Delta_r u\rangle_{\mathcal{V}}}{\|u\|_{\mathcal{V}}^2} = \frac{\|\nabla u\|_{\mathcal{E}}^2}{\|u\|_{\mathcal{V}}^2} = \frac{\frac{1}{2}\sum_{i,j\in V}\omega_{ij}(u_i - u_j)^2}{\sum_{i\in V}d_i^r u_i^2},$$

$$R^+(u) := \frac{\langle u, \Delta_r^+ u\rangle_{\mathcal{V}}}{\|u\|_{\mathcal{V}}^2} = \frac{\|\nabla^+ u\|_{\mathcal{E}}^2}{\|u\|_{\mathcal{V}}^2} = \frac{\frac{1}{2}\sum_{i,j\in V}\omega_{ij}(u_i + u_j)^2}{\sum_{i\in V}d_i^r u_i^2},$$

respectively. By Proposition 1, Δ_r and Δ_r^+ are self-adjoint and positive-semidefinite operators on $\mathcal{V}$, so their eigenvalues will be real and non-negative. The eigenvalues of Δ_r and Δ_r^+ are linked to the extremal values of their Rayleigh quotients by the min-max theorem [19, 26]. In particular, if we denote by $0 \leq \lambda_1 \leq \ldots \leq \lambda_n$ the (possibly repeated) eigenvalues of Δ^+, then $\lambda_1 = \min_{u_1\in\mathcal{V}\setminus\{0\}} R^+(u_1)$ and $\lambda_n = \max_{u_n\in\mathcal{V}\setminus\{0\}} R^+(u_n)$.

In the following proposition we extend a well-known result for the graph Laplacians [58] to include signless graph Laplacians.

Proposition 2. *Let* $r \in [0,1]$. *The following statements are equivalent:*

1. λ *is an eigenvalue of* L_1 *with corresponding eigenvector* v;
2. λ *is an eigenvalue of* L_s *with corresponding eigenvector* $D^{1/2}v$;
3. $2 - \lambda$ *is an eigenvalue of* L_1^+ *with corresponding eigenvector* v;
4. $2 - \lambda$ *is an eigenvalue of* L_s^+ *with corresponding eigenvector* $D^{1/2}v$;
5. λ *and* v *are solutions of the generalized eigenvalue problem* $L_r v = \lambda D^{1-r}v$.

Proof. For $r = 1$ the matrix representations of the graph Laplacian and signless graph Laplacian satisfy $L_1^+ = I + D^{-1}A = 2I - (I - D^{-1}A) = 2I - L_1$. Hence λ is an eigenvalue of L_1 with corresponding eigenvector v if and only if $2 - \lambda$ is an eigenvalue of L_1^+ with the same eigenvector.

Because $L_s = D^{1/2}L_1 D^{-1/2}$, λ is an eigenvalue of L_1 with eigenvector v if and only if λ is an eigenvalue of L_s with eigenvector $D^{1/2}v$. Moreover, since $L_s^+ = 2I - L_s$, we have that $2 - \lambda$ is an eigenvalue of L_s^+ with eigenvector $D^{1/2}v$ if and only if λ is an eigenvalue of L_s with eigenvalue $D^{1/2}v$.

Finally, for $r \in [0,1]$, we have $L_r = D^{1-r}L_1$, hence λ is an eigenvalue of L_1 with corresponding eigenvector v if and only if $L_r v = \lambda D^{1-r}v$. $\square$

Inspired by Proposition 2, we define, for a given graph $G \in \mathcal{G}$ and node subset $S \subset V$, the rescaled indicator function $\tilde{\chi}_S \in \mathcal{V}$, by, for all $j \in V$,

$$(\tilde{\chi}_S)_j := d_j^{\frac{1}{2}} (\chi_S)_j. \tag{7}$$

Proposition 3. *The graph* $G = (V, E, \omega) \in \mathcal{G}$ *has* k *connected components if and only if* $\Delta \in \{\Delta_r, \Delta_s\}$ $(r \in [0,1])$ *has eigenvalue* 0 *with algebraic and geometric multiplicity equal to* k. *In that case, the eigenspace corresponding to the* 0 *eigenvalue is spanned by*

- *the indicator functions* χ_{S_i}, *if* $\Delta = \Delta_r$, *or*
- *the rescaled indicator functions* $\tilde{\chi}_{S_i}$ *(as in* (7)*), if* $\Delta = \Delta_s$.

Here the node subsets $S_i \subset V$, $i \in \{1, \ldots, k\}$, *are such that each connected component of* G *is the subgraph induced by an* S_i.

Proof. We follow the proof in [58]. First we consider the case where $\Delta = \Delta_r$, $r \in [0,1]$. We note that Δ_r is diagonizable in the $\mathcal{V}$ inner product and thus the algebraic multiplicity of any of its eigenvalues is equal to its geometric multiplicity. In this proof we will thus refer to both simply as 'multiplicity'.

For any function $u \in \mathcal{V}$ we have that $\langle u, \Delta_r u \rangle_{\mathcal{V}} = \frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i - u_j)^2$. We have that 0 is an eigenvalue if and only if there exists a $u \in \mathcal{V} \setminus \{0\}$ such that

$$\langle u, \Delta_r u \rangle_{\mathcal{V}} = 0. \tag{8}$$

This condition is satisfied if and only if, for all $i, j \in V$ for which $\omega_{ij} > 0$, $u_i = u_j$.

Now assume that G is connected (hence G has $k = 1$ connected component), then (8) is satisfied if and only if, for all $i, j \in V$, $u_i = u_j$. Therefore any eigenfunction corresponding to the eigenvalue $\lambda_1 = 0$ has to be constant, e.g. $u = \chi_V$. In particular, the multiplicity of λ_1 is 1.

Now assume that G has $k \geq 2$ connected components, let S_i, $i \in \{1, \ldots, k\}$ be the node sets corresponding to the connected components of the graph. Via a suitable

reordering of nodes G will have a graph Laplacian matrix of the form

$$
L_r = \begin{pmatrix}
L_r^{(1)} & 0 & \cdots & 0 \\
0 & L_r^{(2)} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & L_r^{(k)}
\end{pmatrix},
$$

where each matrix $L_r^{(i)}$, $i \in \{1, \ldots, k\}$ corresponds to Δ_r restricted to the connected component induced by S_i. This restriction is itself a graph Laplacian for that component. Because each $L_r^{(i)}$ has eigenvalue zero with multiplicity 1, L_r (and thus Δ_r) has eigenvalue 0 with multiplicity k. We can choose the eigenvectors equal to χ_{S_i} for $i \in \{1, \ldots, k\}$ by a similar argument as in the $k = 1$ case.

Conversely, if Δ_r has eigenvalue 0 with multiplicity k, then G has k connected components, because if G has $l \neq k$ connected components, then by the proof above the eigenvalue 0 has multiplicity $l \neq k$.

For Δ_s we use Proposition 2 to find that the eigenvalues are the same as those of Δ_r, with the corresponding eigenfunctions rescaled as stated in the result. $\qquad \square$

Proposition 4. *Let $G = (V, E, \omega) \in \mathcal{G}$ have k connected components and let the node subsets $S_i \subset V$, $i \in \{1, \ldots, k\}$ be such that each connected component is the subgraph induced by one of the S_i. We denote these subgraphs by G_i. Let $\Delta^+ \in \{\Delta_r^+, \Delta_s^+\}$ ($r \in [0,1]$) and let $0 \leq k' \leq k$. Then Δ^+ has an eigenvalue equal to 0 with algebraic and geometric multiplicity k' if and only if k' of the subgraphs G_i are bipartite. In that case, assume the labelling is such that G_i, $i \in \{1, \ldots, k'\}$ are bipartite with bipartition $(T_i, S_i \setminus T_i)$, where $T_i \subset S_i$. Then the eigenspace corresponding to the 0 eigenvalue is spanned by*

- *the indicator functions $\chi_{T_i} - \chi_{S_i \setminus T_i}$, if $\Delta^+ = \Delta_r^+$, or*
- *the rescaled indicator functions $\tilde{\chi}_{T_i} - \tilde{\chi}_{S_i \setminus T_i}$ (as in (7)), if $\Delta^+ = \Delta_s^+$.*

Proof. First we consider the case where $\Delta^+ = \Delta_r^+$, $r \in [0,1]$. For any vector $u \in \mathcal{V}$ we have that

$$
\langle u, \Delta_r^+ u \rangle_{\mathcal{V}} = \frac{1}{2} \sum_{i,j \in V} \omega_{ij} (u_i + u_j)^2.
$$

Let $k = 1$, then $\lambda_1 = 0$ is an eigenvalue if and only if there exists $u \in \mathcal{V} \setminus \{0\}$ such that $\langle u, \Delta_r^+ u \rangle_{\mathcal{V}} = 0$. This condition is satisfied if and only if, for all $i, j \in V$ for which $\omega_{ij} > 0$ we have

$$
u_i = -u_j. \tag{9}
$$

We claim that this condition in turn is satisfied if and only if G is bipartite. To prove the 'if' part of that claim, assume G is bipartite with bipartition (A, A^c) for some $A \subset V$, and define $u \in \mathcal{V}$ such that $u|_A = -1$ and $u|_{A^c} = 1$. To prove the 'only if' statement, assume G is not bipartite, then there exists an odd cycle in G [12, Theorem 1.4]. Let $i \in V$ be a vertex on this cycle, then by applying condition (9) to all the vertices of the cycle, we find $u_i = -u_i = 0$. Since G is connected, it now follows, by applying condition (9) to all vertices in V, that $u = 0$, which is a contradiction.

The argument above also shows that, if G is bipartite with bipartition (A, A^c), then any eigenfunction corresponding to $\lambda_1 = 0$ is proportional to $u = \chi_A - \chi_{A^c}$. Therefore the eigenvalue 0 has geometric multiplicity 1. Since Δ_r^+ is diagonizable

in the $\mathcal{V}$ inner product the algebraic multiplicity of λ_1 is equal to its geometric multiplicity.

Now let $k \geq 2$ and let S_i, $i \in \{1, \ldots, k\}$ be the node sets corresponding to the connected components of the graph. Via a suitable reordering of nodes the graph G will have a signless Laplacian matrix of the form

$$L_r^+ = \begin{pmatrix} L^{(1)+} & 0 & \cdots & 0 \\ 0 & L^{(2)+} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L^{(k)+} \end{pmatrix},$$

where each matrix $L^{(i)+}$, $i \in \{1, \ldots, k\}$ corresponds to Δ_r^+ restricted to the connected component induced by S_i. This restriction is itself a signless Laplacian for that connected component. Hence, we can apply the case $k = 1$ to each component separately to find that the (algebraic and geometric) multiplicity of the eigenvalue 0 of Δ_r^+ is equal to the number of connected components which are also bipartite.

If G has $k' \leq k$ connected components which are also bipartite, then, without loss of generality, assume that these components correspond to S_i, $i \in \{1, \ldots, k'\}$. Then the corresponding eigenspace is spanned by functions $u^{(i)} = \chi_{T_i} - \chi_{S_i \setminus T_i} \in \mathcal{V}$, $i \in \{1, \ldots, k'\}$, where $T_i \subset S_i$ and $(T_i, S_i \setminus T_i)$ is the bipartition of the bipartite component induced by S_i.

For Δ_s^+ we use Proposition 2 to find the appropriately rescaled eigenfunctions as given in the result. $\square$

Corollary 1. *The eigenvalues of Δ_1, Δ_1^+, Δ_s, and Δ_s^+ are in $[0, 2]$.*

Proof. For Δ_1 the proof can be found in [56, Lemma 2.5]. For completeness we reproduce it here. By Proposition 1 we know that Δ_1 has non-negative eigenvalues. The upper bound is obtained by maximizing the Rayleigh quotient $R(u)$ over all nonzero $u \in \mathcal{V}$. Since $(u_i - u_j)^2 \leq 2(u_i^2 + u_j^2)$ we have that

$$\max_{u \in \mathcal{V} \setminus \{0\}} R(u) = \max_{u \in \mathcal{V} \setminus \{0\}} \frac{\frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i - u_j)^2}{\sum_{i \in V} d_i u_i^2} \leq \max_{u \in \mathcal{V} \setminus \{0\}} \frac{2 \sum_{i \in V} d_i u_i^2}{\sum_{i \in V} d_i u_i^2} = 2.$$

From Proposition 2 it then follows that the eigenvalues of the other operators are in $[0, 2]$ as well. $\square$

3. The Max-Cut problem and Goemans–Williamson algorithm.

3.1. **Maximum cuts.** In order to identify candidate solutions to the Max-Cut problem with node functions in $\mathcal{V}$, we define the subset of binary $\{-1, 1\}$-valued node functions,

$$\mathcal{V}^b := \{u \in \mathcal{V} : \forall\, i \in V, u_i \in \{-1, 1\}\}.$$

For a given function $u \in \mathcal{V}^b$ we define the sets $V_k := \{i \in V, u_i = k\}$ for $k \in \{-1, 1\}$. We say that the partition $C = V_{-1} | V_1$ is the cut induced by u. We define the set of all possible cuts, $\mathcal{C} := \{C : \text{there exists a } u \in \mathcal{V}^b \text{ such that } u \text{ induces the cut } C\}$.

Definition 3.1. Let $G = (V, E, \omega) \in \mathcal{G}$ and let V_1 and V_{-1} be two disjoint subsets of V. The size of the cut $C = V_{-1} | V_1$ is

$$s(C) := \sum_{\substack{i \in V_{-1} \\ j \in V_1}} \omega_{ij}.$$

A maximum cut of G is a cut $C^* \in \mathcal{C}$ such that, for all cuts $C \in \mathcal{C}$, $s(C) \leq s(C^*)$. The size of the maximum cut is

$$\mathrm{mc}(G) := \max_{C \in \mathcal{C}} s(C).$$

Note that if the cut C in Definition 3.1 is induced by $u \in \mathcal{V}^b$, then

$$s(C) = \frac{1}{4}\langle u, \Delta_r u \rangle_{\mathcal{V}}. \tag{10}$$

Moreover, if $C = \emptyset | V_1$ or $C = V_{-1} | \emptyset$, then $s(C) = 0$.

Definition 3.2 (Max-Cut problem)**.** Given a simple, undirected graph $G = (V, E, \omega) \in \mathcal{G}$, find a maximum cut for G.

For a given $G \in \mathcal{G}$ we have $|E| < \infty$, hence a maximum cut for G exists, but note that this maximum cut need not be unique.

The cardinality of the set $\mathcal{V}^b$ is equal to the total number of ways a set of n elements can be partitioned into two disjoint subsets, i.e. $|\mathcal{V}^b| = 2^n$. This highlights the difficulty of finding $\mathrm{mc}(G)$ as n increases. It has been proven that the Max-Cut problem is NP-hard [24]. Obtaining a performance guarantee of $\frac{16}{17}$ or better is also NP-hard [52]. The problem of determining if a cut of a given size exists on a graph is NP-complete [35].

3.2. **The Goemans–Williamson algorithm.** The leading algorithm for polynomial time Max-Cut approximation is the Goemans–Williamson (GW) algorithm [25], which we present in detail below in Algorithm (GW). A problem equivalent to the Max-Cut problem is to find a maximizer which achieves

$$\max_u \frac{1}{2} \sum_{i,j} \omega_{ij}(1 - u_i u_j) \quad \text{subject to } \forall i \in V, u_i \in \{-1, 1\}.$$

The GW algorithm solves a relaxed version of this integer quadratic program, by allowing u to be an n-dimensional vector with unit Euclidean norm. In [25] it is proved that the n-dimensional vector relaxation is an upper bound on the original integer quadratic program. This relaxed problem is equivalent to finding a maximizer which achieves

$$Z_P^* := \max_Y \frac{1}{2} \sum_{i,j \in V, i<j} \omega_{ij}(1 - y_{ij}), \tag{11}$$

where the maximization is over all n by n real positive-semidefinite matrices $Y = (y_{ij})$ with ones on the diagonal. This semidefinite program has an associated dual problem of finding a minimizer which achieves

$$Z_D^* := \frac{1}{2} \sum_{i,j \in V} \omega_{ij} + \frac{1}{4} \min_{\gamma \in \mathbb{R}^n} \sum_{i \in V} \gamma_i, \tag{12}$$

subject to $A + \mathrm{diag}(\gamma)$ being positive-semidefinite, where A is the adjacency matrix of G and $\mathrm{diag}(\gamma)$ is the diagonal matrix with diagonal entries γ_i.

As mentioned in Section 1.1, Algorithm (GW) is proven to have a performance guarantee of 0.878. In Algorithm (GW) below, we use the unit sphere $S_n := \{x \in \mathbb{R}^n : \|x\| = 1\}$, where $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^n$. For vectors $w, \tilde{w} \in \mathbb{R}^n$, $w \cdot \tilde{w}$ denotes the Euclidean inner product.

Other polynomial time Max-Cut approximation algorithms can be found in [15, 53]. Because of the high proven performance guarantee of (GW), we focus

---

Algorithm (GW): The Goemans–Williamson algorithm

Data: The weighted adjacency matrix A of a graph $G \in \mathcal{G}$, and a tolerance ν.

Relaxation step: Use semidefinite programming to find approximate solutions $\tilde{Z}_P^*$ and $\tilde{Z}_D^*$ to (11) and (12), respectively, which satisfy $|\tilde{Z}_P^* - \tilde{Z}_D^*| < \nu$. Use an incomplete Cholesky decomposition on the matrix Y that achieves $\tilde{Z}_P^*$ in (11) to find an approximate solution to

$$w^* \in \operatorname*{argmax}_{w \in (S_n)^n} \frac{1}{2} \sum_{\substack{1 \le i,j, \le n \\ i < j}} \omega_{ij}(1 - w_i \cdot w_j).$$

Hyperplane step: Let $r \in S_n$ be a random vector drawn from the uniform distribution on S_n. Define the cut $C := V_{-1}|V_1$, where

$$V_1 := \{i \in V | w_i \cdot r \ge 0\} \quad \text{and} \quad V_{-1} := V \setminus V_1.$$

---

on comparing our algorithm against it. In [53] the authors use the eigenvector corresponding to the smallest eigenvalue of Δ_0^+, showing that thresholding this eigenvector in a particular way achieves a Max-Cut performance guarantee of $\beta = 0.531$, which with further analysis was improved to $\beta = 0.614$ [51]. Algorithms which provide a solution in polynomial time exist if the graph is planar [30], if the graph is a line graph [29], or if the graph is weakly bipartite [27]. Comparing against [53, 30, 29, 27] is a topic of future research.

4. **Γ-convergence of f_ε^+.** In (2) we introduced the signless Ginzburg–Landau functional $f_\varepsilon^+ : \mathcal{V} \to \mathbb{R}$. In this section we prove minimizers of f_ε^+ converge to solutions of the Max-Cut problem, using the tools of Γ-convergence [13].

We need a concept of convergence in $\mathcal{V}$. Since we can identify $\mathcal{V}$ with $\mathbb{R}^n$ and all norms on $\mathbb{R}^n$ are topologically equivalent, the choice of a particular norm is not of great importance. For definiteness, however, we say that sequence $\{u_k\}_{k \in \mathbb{N}} \subset \mathcal{V}$ converges to a $u_\infty \in \mathcal{V}$ in $\mathcal{V}$ if and only if $\|\hat{u}_k - \hat{u}_\infty\|_\mathcal{V} \to 0$ as $k \to \infty$, where $\hat{u}_k, \hat{u}_\infty \in \mathbb{R}^n$ are the canonical vector representations of u_k, u_∞, respectively.

We will prove that f_ε^+ Γ-converges to the functional $f_0^+ : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$, which is defined as

$$f_0^+(u) := \begin{cases} \sum_{i,j \in V} \omega_{ij}|u_i + u_j|, & \text{if } u \in \mathcal{V}^b, \\ +\infty, & \text{if } u \in \mathcal{V} \setminus \mathcal{V}^b. \end{cases} \tag{13}$$

Lemma 4.1. *Let $G \in \mathcal{G}$. For every $u \in \mathcal{V}^b$, let $C_u \in \mathcal{C}$ be the cut induced by u, then for all $u \in \mathcal{V}$,*

$$f_0^+(u) = \begin{cases} 2\sum_{i,j \in V} \omega_{ij} - 4s(C_u), & \text{if } u \in \mathcal{V}^b, \\ +\infty, & \text{if } u \in \mathcal{V} \setminus \mathcal{V}^b. \end{cases}$$

In particular, if $u^* \in \operatorname*{argmin}_{u \in \mathcal{V}} f_\varepsilon^+(u)$, then $u^* \in \mathcal{V}^b$ and C_{u^*} is a maximum cut of G.

Proof. Because, for $u \in \mathcal{V}^b$, $f_0^+(u) = 2\text{TV}^+(u)$ (with $q = 1$), the result follows by (6). $\square$

Lemma 4.2. *Let $G \in \mathcal{G}$ and $\varepsilon > 0$. There exist minimizers for the functionals $f_\varepsilon^+ : \mathcal{V} \to \mathbb{R}$ and $f_0^+ : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$ from (2) and (13), respectively. Moreover, if $u \in \mathcal{V}$ is a minimizer of f_0^+, then $u \in \mathcal{V}^b$.*

Proof. The potential W satisfies a coercivity condition in the following sense. There exist a $C_1 > 0$ and a C_2 such that, for all $x \in \mathbb{R}$,

$$|x| \geq C_1 \Rightarrow C_2(x^2 - 1) \leq W(x). \tag{14}$$

Combined with the fact that $\|\nabla^+ u\|_{\mathcal{E}} \geq 0$, this shows that f_{ε}^+ is coercive. Since f_{ε}^+ is a (multivariate) polynomial, it is continuous. Thus, by the direct method in the calculus of variations [20, Theorem 1.15] f_{ε}^+ has a minimizer in $\mathcal{V}$.

Since $n \geq 1$, $\mathcal{V}^b \neq \emptyset$ and thus $\inf_{u \in \mathcal{V}} f_0^+(u) < +\infty$. In particular, any minimizer of f_0^+ has to be in $\mathcal{V}^b$. Since $|\mathcal{V}^b| < \infty$ the minimum is achieved. $\square$

Lemma 4.3. *Let $G \in \mathcal{G}$ and let f_{ε}^+ and f_0^+ be as in (2) and (13), respectively. Then f_{ε}^+ Γ-converges to f_0^+ as $\varepsilon \downarrow 0$ in the following sense: If $\{\varepsilon_k\}_{k \in \mathbb{N}}$ is a sequence of positive real numbers such that $\varepsilon_k \downarrow 0$ as $k \to \infty$ and $u_0 \in \mathcal{V}$, then the following lower bound and upper bound conditions are satisfied:*

(LB) for every sequence $\{u_k\}_{k=1}^{\infty} \subset \mathcal{V}$ such that $u_k \to u_0$ as $k \to \infty$, it holds that $f_0^+(u_0) \leq \liminf_{k \to \infty} f_{\varepsilon_k}^+(u_k)$;

(UB) there exists a sequence $\{u_k\}_{k=1}^{\infty} \subset \mathcal{V}$ such that $u_k \to u_0$ as $k \to \infty$ and $f_0^+(u_0) \geq \limsup_{k \to \infty} f_{\varepsilon_k}^+(u_k)$.

Proof. This proof is an adaptation of the proofs in [55, Section 3.1].

Note that

$$f_{\varepsilon}^+(u) = \frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i + u_j)^2 + w_{\varepsilon}(u),$$

where we define $w_{\varepsilon} : \mathcal{V} \to \mathbb{R}$ by

$$w_{\varepsilon}(u) := \frac{1}{\varepsilon} \sum_{i \in V} W(u_i).$$

First we prove that w_{ε} Γ-converges to w_0 as $\varepsilon \downarrow 0$, where

$$w_0(u) := \begin{cases} 0, & \text{if } u \in \mathcal{V}^b, \\ +\infty, & \text{if } u \in \mathcal{V} \setminus \mathcal{V}^b. \end{cases}$$

Let $\{\varepsilon_k\}_{k \in \mathbb{N}}$ be a sequence of positive real numbers such that $\varepsilon_k \downarrow 0$ as $k \to \infty$ and $u_0 \in \mathcal{V}$.

(LB) Note that, for all $u \in \mathcal{V}$ we have $w_{\varepsilon}(u) \geq 0$. Let $\{u_k\}_{k=1}^{\infty}$ be a sequence such that $u_k \to u_0$ as $k \to \infty$. First we assume that $u_0 \in \mathcal{V}^b$, then

$$w_0(u_0) = 0 \leq \liminf_{k \to \infty} w_{\varepsilon_k}(u_k).$$

Next suppose that $u_0 \in \mathcal{V} \setminus \mathcal{V}^b$, then there is an $i \in V$ such that $(u_0)_i \notin \{-1, 1\}$. Since $u_k \to u_0$ as $k \to \infty$, for every $\eta > 0$ there is an $N(\eta) \in \mathbb{N}$ such that for all $k \geq N(\eta)$ we have that $d_i^r |(u_0)_i - (u_k)_i| < \eta$. Define

$$\bar{\eta} := \frac{1}{2} d_i^r \min\{|1 - (u_0)_i|, |-1 - (u_0)_i|\} > 0,$$

then, for all $k \geq N(\bar{\eta})$,

$$|1 - (u_k)_i| \geq \left||1 - (u_0)_i| - |(u_0)_i - (u_k)_i|\right| \geq \frac{1}{2}|1 - (u_0)_i| > 0.$$

Similarly, for all $n \geq N(\bar{\eta})$, $|-1-(u_k)_i| \geq \frac{1}{2}|-1-(u_0)_i| > 0$. Hence, there is a $c > 0$ such that, for all $k \geq N(\bar{\eta})$, $|(u_k)_i| \leq 1 - c$. Thus there is a $C > 0$ such that, for all $k \geq N(\bar{\eta})$, $W((u_k)_i) \geq C$. It follows that

$$\liminf_{k \to \infty} w_{\varepsilon_k}(u_k) \geq \liminf_{k \to \infty} \frac{1}{\varepsilon_k} W((u_k)_i) = \infty = w_0(u_0).$$

(UB) If $u_0 \in \mathcal{V} \backslash \mathcal{V}^b$, then $w_0(u_0)$ and the upper bound condition is trivially satisfied. Now assume $u_0 \in \mathcal{V}^b$. Define the sequence $\{u_k\}_{k=1}^{\infty}$ by, that for all $k \in \mathbb{N}$, $u_k = u_0$. Then, for all $k \in \mathbb{N}$, $w_{\varepsilon_k}(u_k) = 0$ and thus $\limsup_{k \to \infty} w_{\varepsilon_k}(u_k) = 0 = w_0(u_0)$. This concludes the proof that w_{ε} Γ-converges to w_0 as $\varepsilon \downarrow 0$.

It is known that Γ-convergence is stable under continuous perturbations [20, Proposition 6.21], [13, Remark 1.7]; thus $w_{\varepsilon} + p$ Γ-converges to $w_0 + p$ for any continuous $p : \mathcal{V} \to \mathbb{R}$. Since $u \mapsto \frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i + u_j)^2$ is a polynomial and hence a continuous function on $\mathcal{V}$, we find that, as $\varepsilon \downarrow 0$, f_{ε}^+ Γ-converges to $g : \mathcal{V} \to \mathbb{R} \cup \{+\infty\}$, where

$$g(u) := \frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i + u_j)^2 + w_0(u).$$

If $u \in \mathcal{V} \backslash \mathcal{V}^b$, then $g(u) = +\infty$. If $u \in \mathcal{V}^b$, then, for all $i, j \in V$, $(u_i + u_j)^2 = 2|u_i + u_j|$, hence

$$\frac{1}{2} \sum_{i,j \in V} \omega_{ij}(u_i + u_j)^2 = \sum_{i,j \in V} \omega_{ij}|u_i + u_j|.$$

Thus $g = f_0^+$ and the theorem is proven. $\qquad \square$

Lemma 4.4. *Let $G \in \mathcal{G}$ and let f_{ε}^+ be as in (2). Let $\{\varepsilon_k\}_{k \in \mathbb{N}} \subset (0, \infty)$ be a sequence such that $\varepsilon_k \downarrow 0$ as $k \to \infty$, then the sequence $\{f_{\varepsilon_k}^+\}_{k \in \mathbb{N}}$ satisfies the following equi-coerciveness property: If $\{u_k\}_{k \in \mathbb{N}} \subset \mathcal{V}$ is a sequence such that there exists $C > 0$ such that, for all $k \in \mathbb{N}$, $f_{\varepsilon_k}^+(u_k) < C$, then there exists a subsequence $\{u_{k'}\}_{k' \in \mathbb{N}} \subset \{u_k\}_{k \in \mathbb{N}}$ and a $u_0 \in \mathcal{V}^b$ such that $u_{k'} \to u_0$ as $k \to \infty$.*

Proof. This proof closely follows [55, Section 3.1].

From the uniform bound $f_{\varepsilon_k}^+(u_k) < C$, we have that, for all $k \in \mathbb{N}$ and all $i \in V$, $0 \leq W((u_k)_i) \leq C$. Because of the coercivity property (14) of W, the uniform bound on $W((u_k)_i)$ gives, for all $i \in V$, boundedness of $\{d_i^r(u_k)_i^2\}_{k \in \mathbb{N}}$ and thus $\{u_k\}_{k \in \mathbb{N}}$ is bounded in the $\mathcal{V}$-norm. The result now follows by the Bolzano-Weierstrass theorem. $\qquad \square$

With the Γ-convergence and equi-coercivity results from Lemmas 4.3 and 4.4, respectively, in place, we now prove that minimizers of f_{ε}^+ converge to solutions of the Max-Cut problem.

Theorem 4.5. *Let $G \in \mathcal{G}$. Let $\{\varepsilon_k\}_{k \in \mathbb{N}} \subset (0, \infty)$ be a sequence such that $\varepsilon_k \downarrow 0$ as $k \to \infty$ and, for each $k \in \mathbb{N}$, let $f_{\varepsilon_k}^+$ be as in (2) and let u_{ε_k} be a minimizer of $f_{\varepsilon_k}^+$. Then $\min_{u \in \mathcal{V}} f_{\varepsilon_k}^+(u) \to \min_{u \in \mathcal{V}} f_0^+(u)$ as $k \to \infty$. Furthermore, there exists $u_0 \in \mathcal{V}^b$ and a subsequence $\{u_{\varepsilon_{k'}}\}_{k' \in \mathbb{N}} \subset \{u_{\varepsilon_k}\}_{k \in \mathbb{N}}$, such that $\|u_{\varepsilon_{k'}} - u_0\|_{\mathcal{V}} \to 0$ as $k' \to \infty$. Moreover, $u_0 \in \underset{u \in \mathcal{V}}{\mathrm{argmin}}\, f_0^+(u)$, where f_0^+ is as in (13). In particular, if $C_{u_0} \in \mathcal{C}$ is the cut induced by u_0, then C_{u_0} is a maximum cut of G.*

Proof. It is a well-known result from Γ-convergence theory [20, Corollary 7.20], [13, Theorem 1.21] that the equi-coercivity property of $\{f_{\varepsilon_k}\}_{k \in \mathbb{N}}$ from Lemma 4.4

combined with the Γ-convergence property of Lemma 4.3 implies that $\min\limits_{u \in \mathcal{V}} f_{\varepsilon_k}^+(u)$ converge to $\min\limits_{u \in \mathcal{V}} f_0^+(u)$ as $k \to \infty$, and, up to taking a subsequence, minimizers of $f_{\varepsilon_k}^+$ converge to a minimizer of f_0^+ as $k \to \infty$. By Lemma 4.2, if $u_0 \in \operatorname*{argmin}\limits_{u \in \mathcal{V}} f_0^+(u)$, then $u_0 \in \mathcal{V}^b$. By Lemma 4.1, the cut C_{u_0} induced by u_0 is a maximum cut of G. $\qquad\square$

5. The signless MBO algorithm.

5.1. **Algorithm.** One way of attempting to find minimizers of f_ε^+ is via its gradient flow [6]. This is, for example, the method employed in [11] to find approximate minimizers of f_ε. In that case the gradient flow is given by a graph-based analogue of the Allen–Cahn equation [5]. To find the $\mathcal{V}$-gradient flow of f_ε^+ we compute the first variation of the functional f_ε^+: for $t \in \mathbb{R}$, $u, v \in \mathcal{V}$, we have

$$\frac{d}{dt} f_\varepsilon^+(u + tv)|_{t=0} = \langle \Delta_r^+ u, v \rangle_{\mathcal{V}} + \frac{1}{\varepsilon} \langle D^{-r} W' \circ u, v \rangle_{\mathcal{V}},$$

where we used the notation $(D^{-r} W' \circ u)_i = d_i^{-r} W'(u_i)$. This leads to the following $\mathcal{V}$-gradient flow: for all $i \in V$,

$$\begin{cases} \frac{du_i}{dt} = -(\Delta_r^+ u)_i - \frac{1}{\varepsilon} d_i^{-r} W'(u_i), & \text{for } t > 0, \\ u_i = (u_0)_i, & \text{for } t = 0. \end{cases} \tag{15}$$

Since f_ε^+ is not convex, as $t \to \infty$ the solution of the $\mathcal{V}$-gradient flow is not guaranteed to converge to a global minimum, and can get stuck in local minimizers.

In this paper we will not attempt to directly solve the gradient flow equation. That could be the topic of future research. Instead we will use a graph MBO type scheme, which we call the signless MBO algorithm. It is given in (MBO+). Despite there currently not being any rigorous results on the matter, the outcome of this scheme is believed to approximate minimizers of f_ε^+. The original MBO scheme (or threshold dynamics scheme) in the continuum was introduced to approximate motion by mean curvature flow [41, 42]. It consists of iteratively applying (N times) two steps: diffusing a binary initial condition for a time τ and then thresholding the result back to a binary function. In the (suitably scaled) limit $\tau \downarrow 0$, $N \to \infty$, solutions of this process converge to solutions of motion by mean curvature [10]. It is also known that solutions the continuum Allen–Cahn equation (in the limit $\varepsilon \downarrow 0$) converge to solutions of motion by mean curvature [14]. Whether something similar is true for the graph MBO scheme or graph Allen–Cahn equation [56] or something analogous is true for the signless graph MBO scheme are as of yet open questions, but it does suggest that solutions of the MBO scheme (signless MBO scheme) could be closely connected to minimizers of f_ε (f_ε^+). In practice, the graph MBO scheme has proven to be a fast and accurate method for tackling approximate minimization problems of this kind [40, 11].

We see in (MBO+) that in the signless diffusion step the equation that is solved is the gradient flow equation from (15) without the double well potential term. Since we expect the double well potential term in (15) to force the solution to take values close to ± 1, the signless diffusion step in (MBO+) is followed by a thresholding step. Note that, despite our choice of nomenclature, the signless graph 'diffusion' dynamics is expected to be significantly different from standard graph diffusion.

In Figures 1 and 2 we show the minimization of f_ε^+ using (MBO+) with the spectral method (which is explained in Section 5.2) on the AS8 graph and the

---

Algorithm (MBO+): The signless graph MBO algorithm

Data: A signless graph Laplacian $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$ corresponding to a graph $G \in \mathcal{G}$, a signless diffusion time $\tau > 0$, an initial condition $\mu^0 := \chi_{S_0} - \chi_{S_0^c}$ corresponding to a node subset $S_0 \subset V$, a time step dt, and a stopping criterion tolerance η.

Output: A sequence of functions $\{\mu^j\}_{j=0}^N \subset \mathcal{V}^b$ giving the signless MBO evolution of μ^0, a sequence of corresponding cuts $\{C^j\}_{j=0}^N \subset \mathcal{C}$ and their sizes $\{s(C^j)\}_{j=0}^N \subset [0, \infty)$, with largest value s^*.

for $j = 1$ **to** stopping criterion is satisfied, **do**

  Signless diffusion step: Compute $u^*(\tau)$, where $u^* \in \mathcal{V}$ is the solution of the initial value problem

$$\begin{cases} \frac{du(t)}{dt} = -\Delta^+ u(t), & \text{for } t > 0, \\ u(0) = \mu^j. \end{cases} \tag{16}$$

  Threshold step: Define $\mu^j \in \mathcal{V}^b$ by, for $i \in V$,

$$\mu_i^j := T(u_i^*(\tau)) := \begin{cases} 1, & \text{if } u_i^*(\tau) > 0, \\ -1, & \text{if } u_i^*(\tau) \leq 0. \end{cases} \tag{17}$$

  Define the cut $C^j := V_{-1}^j | V_1^j$, where $V_{\pm 1}^j := \{i \in V : \mu_i^j = \pm 1\}$ and compute $s(C^j)$.

  Set $N = j$.

  if $\frac{\|\mu^j - \mu^{j-1}\|_2^2}{\|\mu^j\|_2^2} < \eta$ **then**

    ⌊ Stop

Find the largest cut size: Set $s^* := \max_{1 \leq j \leq N} s(C^j)$.

GNutella09 graph (see Section 7.3). The (MBO+) iteration numbers j are indicated along the x-axis. The y-axis shows the value of $f_\varepsilon^+(\mu^j)$. What we see in both figures is that the overall tendency is for the (MBO+) algorithm to decrease the value of $f_\varepsilon^+(\mu^j)$, however, in some iterations the value increases. This is why in (MBO+) we output the cut size which is largest among all iterations computed and use that as the final output, if it outperforms the cut C which (MBO+) returns. Alternatively, in order to save on computing memory, one could also keep track of the largest cut size found so far in each iteration and discard the other cut sizes, or accept the final cut size $s(C^N)$ as approximation to s^*. The result we report in this paper are all based on the output s^*.

In our experiments we choose the stopping criterion tolerance $\eta = 10^{-8}$.

5.2. **Spectral decomposition method.** In this paper we will compare two implementations of the (MBO+) algorithm, which differ in the way they solve (16) for $t \in [0, \tau]$. In the next section we consider an explicit Euler method, but first we discuss a spectral decomposition method. In order to solve (16) we use spectral decomposition of the signless graph Laplacian $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$. Let $\lambda_k \geq 0$, $k \in \{1, \ldots, n\}$ be the eigenvalues of Δ^+. We assume $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_n$ and list eigenvalues multiple times according to their multiplicity. Let $\phi^k \in \mathcal{V}$ be an eigenfunction corresponding to λ_k, chosen such that $\{\phi_k\}_{k=1}^n$ is a set of orthonormal functions in $\mathcal{V}$. We then use the decomposition
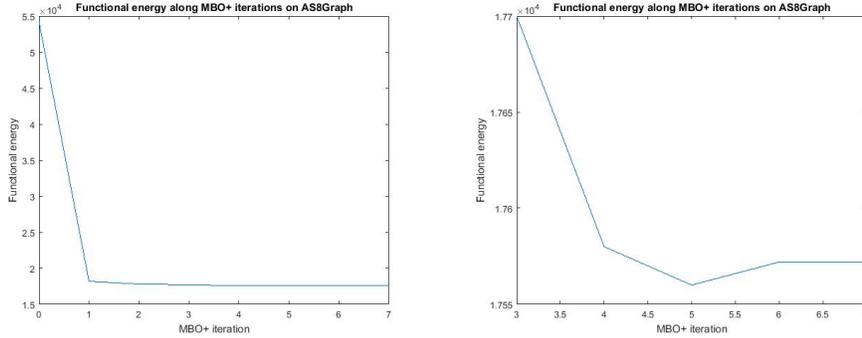
FIGURE 1. The value $f_\varepsilon^+(\mu^j)$ as a function of the iteration number j in the (MBO+) scheme on AS8Graph, using the spectral method and Δ_1^+, with $K = 100$, and $\tau = 20$. The left hand plot shows the initial condition and all iterations of the (MBO+) scheme on AS8Graph, where as the right hand plot displays the 3rd to the final iterations of the (MBO+) scheme on AS8Graph.
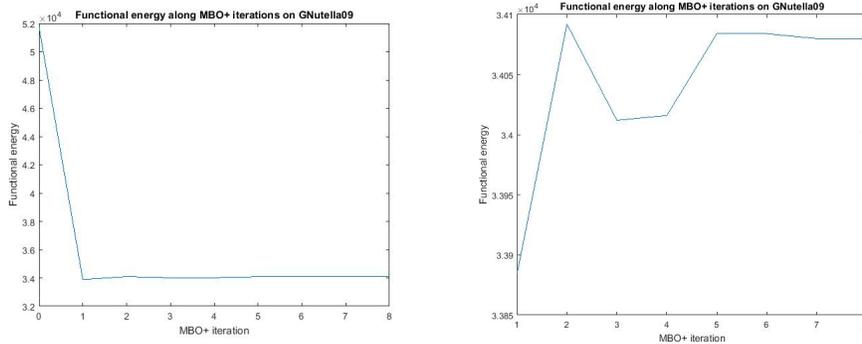

FIGURE 2. The value $f_\varepsilon^+(\mu^j)$ as a function of the iteration number j in the (MBO+) scheme on the GNutella09 graph, using the spectral method and Δ_1^+, with $K = 100$, and $\tau = 20$. The left hand plot shows the initial condition and all iterations of the (MBO+) scheme on GNutella09, where as the right hand plot displays all iterations of the (MBO+) scheme on GNutella09, without the initial condition.

$$u^*(\tau) = \sum_{k=1}^{n} e^{-\lambda_k \tau} \langle \phi^k, u(0) \rangle_\mathcal{V} \, \phi^k \tag{18}$$

to solve (16).

For Δ_s^+ we use the Euclidean inner product instead of the $\mathcal{V}$ inner product in (18), because the Laplacian Δ_s^+ is not of the form as given in (4). The optimal choice for τ with respect to the cut size obtained by (MBO+) is a topic for future research. Based on trial and error, we decided to use $\tau = 20$ in the results we present in Section 7, when using Δ_1^+ or Δ_s^+ as our operator. We use $\tau = \frac{40}{\lambda_n}$ when

using Δ_0^+ as our operator, where λ_n is the largest eigenvalue of Δ_0^+. The division of τ by half of the largest eigenvalue of Δ_0^+ is justified in Section 5.4. In Section 8 we investigate how cut sizes change with varying τ.

A computational advantage of the spectral decomposition method is that we do not necessarily need to use all of the eigenvalues and eigenfunctions of the signless Laplacian. We can use only the K eigenfunctions corresponding to the smallest eigenvalues in our decomposition (18). To be explicit, doing this replaces n in (18) by K. In Section 8 we show how increasing K beyond a certain point has little effect on the size of the cut obtained by (MBO+) for three examples. We refer to using the K eigenfunctions corresponding to the smallest eigenvalues in the decomposition as spectral truncation.

By Proposition 2, we can compute the K smallest eigenvalues λ_k ($k \in \{1, \dots, K\}$) of Δ_1^+ and Δ_s^+ by first computing the K largest eigenvalues $\hat{\lambda}_l$ ($l \in \{n-K+1, \dots, n\}$) of Δ_1 and Δ_s respectively instead and then setting $\lambda_k = 2 - \hat{\lambda}_{n-k+1}$. There is not a similar property for Δ_0^+ however. Proving upper bounds on the largest eigenvalues of Δ_0 and Δ_0^+ is an active area of research. [28, 49, 59].

We use the MATLAB `eigs` function to calculate the K eigenpairs of the signless Laplacian. This function [39] uses the Implicitly Restarted Arnoldi Method (IRAM) [50], which can efficiently compute the largest eigenvalues and corresponding eigenvectors of sparse matrices. The function `eigs` firstly computes the orthogonal projection of the matrix you want eigenpairs from, and a random vector, onto the matrix's K-dimensional Krylov subspace. This projection is represented by a smaller $K \times K$ matrix. Then `eigs` calculates the eigenvalues of this $K \times K$ matrix, whose eigenvalues are called Ritz eigenvalues. The Ritz eigenvalues are computed efficiently using a QR method [23]. Computationally these Ritz eigenvalues typically approximate the largest eigenvalues of the original matrix. The time complexity of IRAM is currently unknown, but in practice it produces approximate eigenpairs efficiently.

If the matrix of which the eigenvalues are to be computed is symmetric, the MATLAB `eigs` function simplifies to the Implicitly Restarted Lanczos Method (IRLM) [17], therefore typically in practice `eigs` will usually compute the eigenvalues and eigenfunctions of Δ_s^+ faster than those of Δ_1^+.

Using the IRLM for computing the eigenpairs of Δ_0^+ corresponding to its smallest eigenvalues is inefficient. In our experiments using the MATLAB `eig` function to calculate all eigenpairs of Δ_0^+ and choosing the K eigenpairs corresponding to the smallest eigenvalues for the decomposition (18) was faster than using the IRLM to calculate the K eigenpairs of Δ_0^+. Hence, the results discussed in this paper are obtained with `eig` when using Δ_0^+ and `eigs` when using Δ_1^+ or Δ_s^+.

If we use the MATLAB `eigs` function when using our spectral decomposition method we cannot a priori determine the time complexity for (MBO+), because practical experiments have shown the complexity of the IRAM and IRLM methods is heavily dependent on the matrix to which they are applied [47]. If we choose to use the MATLAB `eig` function then the time complexity of (MBO+) is $\mathcal{O}(n^3)$, which is the time complexity of computing all eigenpairs of an $n \times n$ matrix. All other remaining steps of (MBO+) require fewer operations to compute.

5.3. **Explicit Euler method.** We also compute the solution of (16) for $t \in [0, \tau]$ using an explicit finite difference scheme,

$$\begin{cases} u^{m+1} = u^m - \Delta^+ u^m dt, & \text{for } m \in \{0, 1, \ldots, M\}, \\ u^0 = u(0) \end{cases} \tag{19}$$

for the same choice of τ as in (18). For $M \in \mathbb{N}$, $dt = \frac{\tau}{M}$, and we set $u^*(\tau) = u^M$.

If $G \in \mathcal{G}$ then (MBO+) using the Euler method will have a time complexity of $\mathcal{O}(|E|)$, because of the sparsity of the signless Laplacian matrix. When zero entries are ignored, the multiplication of the vector u^m by Δ^+ takes $4|E| + 2n$ operations to compute. Since $G \in \mathcal{G}$ has no isolated nodes, $|E| \geq n - 1$, therefore, when n is large enough, $4|E| > 2n$ and hence the time complexity of the multiplication is $\mathcal{O}(|E|)$. All other remaining steps in (MBO+) using the Euler method require fewer operations to compute.

In Section 8.3 we show some results for (MBO+) when solving (16) using an implicit finite difference scheme, comparing against the results of (MBO+) obtained using (19) to solve (16).

5.4. **(MBO+) pinning condition.** For (MBO+) we have that choosing τ too small causes trivial dynamics in the sense that, for any j, $\mu^j = \mu^0$ in (MBO+). In this section we prove a result which shows that such a τ is inversely proportional to the largest eigenvalue of the signless Laplacian chosen for (MBO+).

We define $d_- := \min_{i \in V} d_i$, and $d_+ := \max_{i \in V} d_i$. Let $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$, then the operator norm $\|\Delta^+\|_\mathcal{V}$ is defined by

$$\|\Delta^+\|_\mathcal{V} := \sup_{u \in \mathcal{V} \setminus \{0\}} \frac{\|\Delta^+ u\|_\mathcal{V}}{\|u\|_\mathcal{V}}$$

We define the maximum norm of $\mathcal{V}$ by $\|u\|_{\mathcal{V},\infty} := \max\{|u_i| : i \in V\}$.

Lemma 5.1. *Let $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$. The operator norm $\|\Delta^+\|_\mathcal{V}$ and the largest eigenvalue λ_n of Δ^+ are equal. This implies that, for all $u \in \mathcal{V}$,*

$$\|\Delta^+ u\|_\mathcal{V} \leq \lambda_n \|u\|_\mathcal{V}.$$

Proof. See [56, Lemma 2.5]. $\qquad\qquad\square$

Lemma 5.2. *The norms $\| \cdot \|_\mathcal{V}$ and $\| \cdot \|_{\mathcal{V},\infty}$ are equivalent, with optimal constants given by*

$$d_-^{\frac{r}{2}} \|u\|_{\mathcal{V},\infty} \leq \|u\|_\mathcal{V} \leq \|\chi_V\|_\mathcal{V} \|u\|_{\mathcal{V},\infty}.$$

Proof. See [56, Lemma 2.2]. $\qquad\qquad\square$

Theorem 5.3. *Let $G \in \mathcal{G}$, and let λ_n be the largest eigenvalue of the signless Laplacian $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$. Let $S_0 \subset V$, $\mu^0 := \chi_{S_0} - \chi_{S_0^c}$, and let $\mu^1 \in \mathcal{V}^b$ be the result of applying one (MBO+) iteration to μ^0. If*

$$\tau < \lambda_n^{-1} \log(1 + d_-^{\frac{r}{2}} \|\chi_V\|_\mathcal{V}^{-1}),$$

then $\mu^1 = \mu^0$.

Proof. This proof closely follows the proof of a similar result in [56, Section 4.2].

If $\|e^{-\tau \Delta^+} \mu^0 - \mu^0\|_{\mathcal{V},\infty} < 1$, then $\mu^1 = \mu^0$. Using Lemma 5.2, we compute

$$\|e^{-\tau \Delta^+} \mu^0 - \mu^0\|_{\mathcal{V},\infty} \leq d_-^{-\frac{r}{2}} \|e^{-\tau \Delta^+} \mu^0 - \mu^0\|_\mathcal{V} \leq d_-^{-\frac{r}{2}} \|e^{-\tau \Delta^+} - \text{Id}\|_\mathcal{V} \|\mu^0\|_\mathcal{V}.$$

Moreover, since $\langle \chi_{S_0}, \chi_{S_0^c} \rangle_{\mathcal{V}} = 0$, we have $\|\mu^0\|_{\mathcal{V}}^2 = \|\chi_{S_0}\|_{\mathcal{V}}^2 + \|\chi_{S_0^c}\|_{\mathcal{V}}^2 = \|\chi_{S_0} + \chi_{S_0^c}\|_{\mathcal{V}}^2 = \|\chi_V\|_{\mathcal{V}}^2$.

Using the triangle inequality and the submultiplicative property (see [48] for example) of $\|\cdot\|_{\mathcal{V}}$, we compute $\|e^{-\tau \Delta^+} - \mathrm{Id}\|_{\mathcal{V}} \leq \sum_{k=1}^\infty \frac{1}{k!} (\tau \|\Delta^+\|_{\mathcal{V}})^k = e^{\lambda_n \tau} - 1$. Therefore, if $\tau < \lambda_n^{-1} \log(1 + d_-^{\frac{r}{2}} \|\chi_V\|^{-1})$, then $\mu^1 = \mu^0$. $\qquad\square$

As stated in Section 5.2, we choose $\tau = 20$ as diffusion time for (MBO+) using Δ_1^+ or Δ_s^+, and $\tau = \frac{40}{\lambda_n}$ when using (MBO+) with Δ_0^+ as the choice of operator. This is due to $\tau = 20$ often being too large when using (MBO+) with Δ_0^+. Choosing $\tau = 20$ for (MBO+) using Δ_0^+ causes the solution to converge to $u(\tau) = 0$ to machine precision. We therefore choose $\tau = \frac{40}{\lambda_n}$ for Δ_0^+ since 5.3 implies a suitable choice of τ for (MBO+) with respect to obtaining non-trivial output cuts is inversely proportional to the largest eigenvalue of the chosen operator Δ^+. Since $\lambda_n = 2$ for Δ_1^+ and Δ_s^+ we choose to divide τ by $\frac{\lambda_n}{2}$ for Δ_0^+.

6. Γ-convergence of the Lyapunov functional for (MBO+) evolutions.

6.1. **Lyapunov functional.** In this section, in (20), we introduce a Lyapunov functional for (MBO+). We prove that, when properly rescaled, this functional Γ-converges to the signless total variation. In our setup and proofs we follow earlier similar work in [56, 57] which in turn was inspired by results in the continuum setting obtained in [22]. We first introduce some needed notation.

Define the space of functions $\mathcal{K} := \{u \in \mathcal{V} : \forall i \in V, u_i \in [-1, 1]\}$. In analogy with [56] we use our operators to define a graph curvature using our signless definitions. If $C = S | S^c$ we define the signless normal of a set by

$$\nu_{ij}^S := \mathrm{sgn}((\nabla^+ (\chi_S - \chi_{S^c})_{ij}) = \begin{cases} 1, & \text{if } i, j \in S, \\ -1, & \text{if } i, j \in S^c, \\ 0, & \text{otherwise,} \end{cases}$$

and our curvature as

$$(\kappa_S^{q,r,+})_i := (\mathrm{div}^+ \nu^S)_i = d_i^{-r} \begin{cases} \sum_{j \in S} \omega_{ij}^q, & \text{if } i \in S, \\ -\sum_{j \in S^c} \omega_{ij}^q, & \text{if } i \in S^c. \end{cases}$$

By direct computation we note that $\Delta^+ (\chi_S - \chi_{S^c})_i = 2\kappa_S^{1,r,+}$. We define the volume of a node set $S \subseteq V$ by

$$\mathrm{Vol}\, S := \|\chi_S\|_{\mathcal{V}}^2 = \sum_{i \in S} d_i^r.$$

Proposition 5. *Let $G \in \mathcal{G}$ and for $r \in [0, 1]$, let $\Delta^+ = \Delta_r^+$. Then, for all $u \in \mathcal{V} \setminus \{0\}$ we have that*

$$\langle u, e^{-\tau \Delta^+} u \rangle_{\mathcal{V}} = \sum_{i \in V} e^{-\tau \lambda_i} \langle u, \phi^i \rangle_{\mathcal{V}}^2 > 0.$$

Proof. This has been proved in [57] using the operator Δ instead of Δ^+. Our proof here, which we include for completeness, is similar. We expand u on the $\mathcal{V}$-orthonormal basis formed by the eigenfunctions $\{\phi^i\}_{i=1}^n$:

$$\langle u, e^{-\tau \Delta^+} u \rangle_{\mathcal{V}} = \left\langle \sum_{i \in V} \langle u, \phi^i \rangle_{\mathcal{V}} \, \phi^i, \sum_{j \in V} e^{-\tau \lambda_j} \langle u, \phi^j \rangle_{\mathcal{V}} \, \phi^j \right\rangle_{\mathcal{V}}$$

$$= \sum_{i,j \in V} e^{-\tau \lambda_j} \langle u, \phi^i \rangle_{\mathcal{V}} \langle u, \phi^j \rangle_{\mathcal{V}} \, \delta_{ij} = \sum_{i \in V} e^{-\tau \lambda_i} \langle u, \phi^i \rangle_{\mathcal{V}}^2,$$

where δ_{ij} is the Kronecker delta function. This expression is strictly positive since u is not the zero function. $\qquad \square$

Proposition 6. *Let $G \in \mathcal{G}$ and let $C \in \mathcal{C}$. Then for $S \subseteq V$ we have that*

$$\mathrm{TV}^+(\chi_S - \chi_{S^c}) = \langle \kappa_S^{1,r,+}, \chi_S - \chi_{S^c} \rangle_{\mathcal{V}}.$$

Proof. By direct computation we show that

$$\langle \kappa_S^{q,r,+}, \chi_S - \chi_{S^c} \rangle_{\mathcal{V}} = d_i^r \sum_{i \in S} \left(\sum_{j \in S} d_i^{-r} \omega_{ij}^q \right) - d_i^r \sum_{i \in S^c} \left(- \sum_{j \in S^c} d_i^{-r} \omega_{ij}^q \right)$$

$$= \sum_{i,j \in S} \omega_{ij}^q + \sum_{i,j \in S^c} \omega_{ij}^q.$$

By setting $q = 1$ we see that if $C = S | S^c$, then

$$\sum_{i,j \in S} \omega_{ij}^q + \sum_{i,j \in S^c} \omega_{ij}^q = \sum_{i,j \in V} \omega_{ij} - 2 \left(\sum_{i \in S, j \in S^c} \omega_{ij} \right) = \sum_{i,j \in V} \omega_{ij} - 2s(C) = \mathrm{TV}^+(\chi_S - \chi_{S^c}).$$

$\square$

Proposition 7. *Let $G \in \mathcal{G}$ and, for $r \in [0,1]$, let $\Delta^+ = \Delta_r^+$. Let (λ_i, ϕ^i) ($i \in \{1, \ldots, n\}$) denote the eigenpairs of Δ^+, where the eigenfunctions are $\mathcal{V}$-orthonormal. Then for $S \subseteq V$ we have that*

$$\mathrm{TV}^+(\chi_S - \chi_{S^c}) = \frac{1}{2} \sum_{i \in V} \lambda_i \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}}^2.$$

Proof. By Proposition 6 we have

$$\mathrm{TV}^+(\chi_S - \chi_{S^c}) = \langle \kappa_S^{1,r,+}, \chi_S - \chi_{S^c} \rangle_{\mathcal{V}},$$

which can be expressed as

$$\left\langle \kappa_S^{1,r,+}, \sum_{i \in V} \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}} \, \phi^i \right\rangle_{\mathcal{V}} = \sum_{i \in V} \langle \kappa_S^{1,r,+}, \phi^i \rangle_{\mathcal{V}} \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}}.$$

We have that $\Delta^+(\chi_S - \chi_{S^c}) = 2\kappa_S^{1,r,+}$, therefore

$$\mathrm{TV}^+(\chi_S - \chi_{S^c}) = \frac{1}{2} \sum_{i \in V} \langle \Delta^+(\chi_S - \chi_{S^c}), \phi^i \rangle_{\mathcal{V}} \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}}.$$

Since Δ^+ is self-adjoint in the $\mathcal{V}$ inner product by Proposition 1, we have that

$$\frac{1}{2} \sum_{i \in V} \langle \Delta^+(\chi_S - \chi_{S^c}), \phi^i \rangle_{\mathcal{V}} \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}}$$

$$= \frac{1}{2} \sum_{i \in V} \langle \chi_S - \chi_{S^c}, \Delta^+ \phi^i \rangle_{\mathcal{V}} \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}}$$

$$= \frac{1}{2} \sum_{i \in V} \langle \chi_S - \chi_{S^c}, \lambda_i \phi^i \rangle_{\mathcal{V}} \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}} = \frac{1}{2} \sum_{i \in V} \lambda_i \langle \chi_S - \chi_{S^c}, \phi^i \rangle_{\mathcal{V}}^2.$$

$\square$

Lemma 6.1. *Let $G \in \mathcal{G}$, $\tau > 0$, and for $r \in [0,1]$, let $\Delta^+ = \Delta_r^+$. Define $J_\tau : \mathcal{V} \to \mathbb{R}$ by*

$$J_\tau(u) := -\langle u, e^{-\tau \Delta^+} u \rangle_{\mathcal{V}} + \operatorname{Vol} V. \tag{20}$$

Then the following hold.

1. *The functional $J_\tau(u)$ is strictly concave and Fréchet differentiable.*
2. *Its derivative at $u \in \mathcal{V}$ in the direction of $v \in \mathcal{V}$ is given by*

$$dJ_\tau^u(v) = -2\langle e^{-\tau \Delta^+} u, v \rangle_{\mathcal{V}}.$$

3. *Let $\mu^0 := \chi_{S_0} - \chi_{S_0^c}$ for some $S_0 \subset V$ and let $\{\mu^j\}_{j=1}^N$ be a sequence generated by (MBO+). For all $j \in \{1, \ldots, N\}$, μ^j minimizes $dJ_\tau^{\mu^{j-1}}(\cdot)$ over $\mathcal{K}$:*

$$\mu^j \in \operatorname*{argmin}_{v \in \mathcal{K}} dJ_\tau^{\mu^{j-1}}(v). \tag{21}$$

4. *For all $j \in \{1, \ldots, N\}$, $J_\tau(\mu^j) \leq J_\tau(\mu^{j-1})$, with equality attained if and only if $\mu^j = \mu^{j-1}$.*

Remark 1. The final property in the lemma above shows that J_τ is a Lyapunov functional for (MBO+). All the results in Lemma 6.1 remain true if a constant term is added to J_τ. In particular, they remain true if the term $\operatorname{Vol} V$ is removed from (20). We included this term in the definition of J_τ, however, because it is needed in Section 6.2 for our Γ-convergence results.

Proof of Lemma 6.1. In [56, Lemma 4.5, Proposition 4.6] and [57, Lemma 5.5] the authors derive similar Lyapunov functionals for two graph based MBO schemes similar to (MBO+). This proof is similar to those proofs.

We first compute the Gateaux derivative at $u \in \mathcal{V}$ in the direction of $v \in \mathcal{V}$. Let $\varepsilon \in \mathbb{R}$, then

$$dJ_\tau^u(v) := \frac{dJ_\tau(u + \varepsilon v)}{d\varepsilon}\Big|_{\varepsilon = 0} = -2\langle e^{-\tau \Delta^+} u, v \rangle_{\mathcal{V}},$$

where we have applied the fact that Δ^+, and thus also $e^{-\tau \Delta^+}$ are self-adjoint. Since $u \mapsto dJ_\tau^u(v)$ is continuous, J_τ is also Fréchet differentiable. Moreover, if $v \in \mathcal{V} \setminus \{0\}$ the second directional derivative is given by

$$\frac{d^2 J_\tau(u + \varepsilon v)}{d\varepsilon^2}\Big|_{\varepsilon = 0} = -2\langle v, e^{-\tau \Delta^+} v \rangle_{\mathcal{V}}$$

which, by Proposition 5, is strictly negative. Therefore J_τ is strictly concave.

To prove the third property, let $j \in \{1, \ldots, N\}$. Then μ^j is generated by (MBO+) and thus it takes values in $\{-1, 1\}$. Let $\mu^* \in \mathcal{K} \setminus \{\mu^j\}$ and let $i \in V$ be such that $\mu_i^* \neq \mu_i^j$. Assume first that $\mu_i^j = 1$ and hence $\mu_i^* \in [-1, 1)$. Then, by (16), (17), we have $\left(e^{-\tau \Delta^+} \mu^{j-1}\right)_i > 0$ and thus $-2\left(e^{-\tau \Delta^+} \mu^{j-1}\right)_i d_i^r \mu_i^j < -2\left(e^{-\tau \Delta^+} \mu^{j-1}\right)_i d_i^r \mu_i^*$. If, on the other hand $i \in V$ is such that $\mu_i^j = -1$ and hence $\mu_i^* \in (-1, 1]$, then $\left(e^{-\tau \Delta^+} \mu^{j-1}\right)_i \leq 0$ and thus $-2\left(e^{-\tau \Delta^+} \mu_i^{j-1} d_i^r \mu^j\right)_i \leq -2\left(e^{-\tau \Delta^+} \mu^{j-1}\right)_i d_i^r \mu_i^*$. Hence, for all $\mu^* \in \mathcal{K} \setminus \{\mu^j\}$, we have $dJ_\tau^{\mu^{j-1}}(\mu^j) \leq dJ_\tau^{\mu^{j-1}}(\mu^*)$ and therefore μ^j is a minimizer.

Since J_τ is strictly concave and $dJ_\tau^{\mu^{j-1}}$ is linear we conclude that if $\mu^j \neq \mu^{j+1}$, then

$$J_\tau(\mu^{j+1}) - J_\tau(\mu^j) < dJ_\tau^{\mu^j}(\mu^{j+1} - \mu^j) = dJ_\tau^{\mu^j}(\mu^{j+1}) - dJ_\tau^{\mu^j}(\mu^j) \leq 0.$$

The inequality follows from (21). If $\mu^{j+1} = \mu^j$ then $J_\tau(\mu^{j+1}) - J_\tau(\mu^j) = 0$. $\square$

Remark 2. Our proof of (21) above also shows that minimizers in (21) need not be unique: at vertices $i \in V$ where $\left(e^{-\tau \Delta^+} \mu^{j-1} \right)_i \neq 0$ any minimizer has to agree with μ^j obtained by (MBO+), but at vertices i where $\left(e^{-\tau \Delta^+} \mu^{j-1} \right)_i = 0$ minimizers can take any value in $[-1, 1]$.

6.2. **Γ-convergence proof.** Let $r \in [0, 1]$ and $\Delta^+ = \Delta_r^+$. For $\tau > 0$ we define a rescaled version of J_τ, restricted to $\mathcal{K}$: define $\hat{J}_\tau : \mathcal{K} \to \mathbb{R} \cup \{\infty\}$ by $\hat{J}_\tau := \frac{1}{\tau} J_\tau|_{\mathcal{K}}$, i.e.

$$\hat{J}_\tau(u) := -\frac{1}{\tau} \langle u, e^{-\tau \Delta^+} u \rangle_{\mathcal{V}} + \frac{1}{\tau} \mathrm{Vol} V. \tag{22}$$

We prove that for $\tau \downarrow 0$ this functional Γ-converges to $\hat{J}_0 : \mathcal{K} \to \mathbb{R} \cup \{\infty\}$ defined by

$$\hat{J}_0(u) := \begin{cases} 2\mathrm{TV}^+(u), & \text{if } u \in \mathcal{V}^b, \\ +\infty, & \text{if } u \in \mathcal{K} \backslash \mathcal{V}^b, \end{cases} \tag{23}$$

where $q = 1$. Note that $\hat{J}_0 = f_0^+|_{\mathcal{K}}$, where f_0^+ is as in Lemma 4.1. We use the same notion of convergence in $\mathcal{V}$ is the same as in Section 4.

Lemma 6.2. *Let $G \in \mathcal{G}$ and let $\hat{J}_\tau$ and $\hat{J}_0$ be as in (22) and (23), respectively. Then $\hat{J}_\tau$ Γ-converges to $\hat{J}_0$ as $\tau \downarrow 0$ in the following sense: If $\{\tau_k\}_{k \in \mathbb{N}}$ is a sequence of positive real numbers such that $\tau_k \downarrow 0$ as $k \to \infty$ and $u_0 \in \mathcal{K}$, then the following lower bound and upper bound conditions are satisfied:*

(LB) for every sequence $\{u_k\}_{k=1}^\infty \subset \mathcal{K}$ such that $u_k \to u_0$ as $k \to \infty$, it holds that
$$\hat{J}_0(u_0) \leq \liminf_{k \to \infty} \hat{J}_{\tau_k}(u_k);$$
(UB) there exists a sequence $\{u_k\}_{k=1}^\infty \subset \mathcal{K}$ such that $u_k \to u_0$ as $k \to \infty$ and
$$\hat{J}_0(u_0) \geq \limsup_{k \to \infty} \hat{J}_{\tau_k}(u_k).$$

Proof. (LB) By Proposition 5 we can express (22) as

$$\hat{J}_\tau(u) = -\frac{1}{\tau} \sum_{i \in V} e^{-\tau \lambda_i} \langle u, \phi^i \rangle_{\mathcal{V}}^2 + \frac{1}{\tau} \mathrm{Vol}\, V$$

$$= -\sum_{i \in V} \frac{e^{-\tau \lambda_i} - 1}{\tau} \langle u, \phi^i \rangle_{\mathcal{V}}^2 - \frac{1}{\tau} \sum_{i \in V} \langle u, \phi^i \rangle_{\mathcal{V}}^2 + \frac{1}{\tau} \mathrm{Vol}\, V.$$

Since $\tau_k \downarrow 0$ as $k \to \infty$, we have $\dfrac{e^{-\tau_k \lambda_i} - 1}{\tau_k} \to \dfrac{d}{dt} e^{-t\lambda_i} \Big|_{t=0}$ as $k \to \infty$. Moreover, since $u_k \to u_0$ as $k \to \infty$, we deduce that, for all $i \in V$, $-\lim\limits_{k \to \infty} \dfrac{e^{-\tau_k \lambda_i} - 1}{\tau_k} \langle u_k, \phi^i \rangle_{\mathcal{V}}^2 = \lambda_i \langle u_0, \phi^i \rangle_{\mathcal{V}}^2$. It follows that

$$\liminf_{k \to \infty} \hat{J}_\tau(u_k) = \sum_{i \in V} \lambda_i \langle u_0, \phi^i \rangle_{\mathcal{V}}^2 + \liminf_{k \to \infty} \frac{1}{\tau_k} \left(\mathrm{vol}\, V - \sum_{i \in V} \langle u_k, \phi^i \rangle_{\mathcal{V}}^2 \right). \tag{24}$$

Since $u_k \in \mathcal{K}$, we have, for all $i \in V$, $(u_k)_i^2 \leq 1$ and thus

$$\sum_{i \in V} \langle u_k, \phi^i \rangle_{\mathcal{V}}^2 = \langle u_k^2, \chi_V \rangle_{\mathcal{V}} \leq \langle \chi_V, \chi_V \rangle_{\mathcal{V}} = \mathrm{Vol}\, V. \tag{25}$$

For the first equality we have used [57, Lemma 4.6] (which in its original form is formulated for eigenfunctions of the graph Laplacian, but holds without further change for any functions ϕ^i which form an orthonormal basis of $\mathcal{V}$). Hence

$$\liminf_{k\to\infty}\frac{1}{\tau_k}\left(\text{vol}\,V-\sum_{i\in V}\langle u_k,\phi^i\rangle_{\mathcal{V}}^2\right)\geq 0.$$

Now assume that $u_0\in\mathcal{V}^b$, then $u_0=\chi_S-\chi_{S^c}$ for some $S\subseteq V$. It follows from the above, combined with Proposition 7, that

$$\liminf_{k\to\infty}\hat{J}_\tau(u_k)\geq\sum_{i\in V}\lambda_i\langle u_0,\phi^i\rangle_{\mathcal{V}}^2=2\text{TV}^+(u_0)=\hat{J}_0(u_0).$$

Next assume that $u_0\in\mathcal{K}\backslash\mathcal{V}^b$. Then there is a $j\in V$ such that $(u_0)_j\in(-1,1)$. Since $u_k\to u_0$ as $k\to\infty$, there exists an $\eta\in(0,1)$ such that, for all k large enough, $(u_k)_j^2\in[0,1-\eta]$. Using the fact that, for all $i\in V$, we have $d_i^r\big(1-(u_k)_i^2\big)\geq 0$, we find that, for large k,

$$\text{vol}\,V-\sum_{i\in V}\langle u_k,\phi^i\rangle_{\mathcal{V}}^2=\sum_{i\in V}\big(d_i^r-d_i^r(u_k)_i^2\big)\geq d_j^r\big(1-(u_k)_j^2\big)\geq d_j^r\big(1-(1-\eta)\big)=\eta d_j^r>0.$$

Therefore, using (24), we have

$$\liminf_{k\to\infty}\hat{J}_{\tau_k}(u_k)\geq\liminf_{k\to\infty}\frac{1}{\tau_k}d_j^r\eta=+\infty=\hat{J}_0(u_0).$$

(UB) If $u_0\in\mathcal{K}\backslash\mathcal{V}^b$ then the upper bound is trivially satisfied. If $u_0\in\mathcal{V}^b$, then we define our recovery sequence $\{u_k\}$ such that, for all $k\in\mathbb{N}$, $u_k:=u_0$. As in the proof of (LB) above, we have that

$$\limsup_{k\to\infty}\hat{J}_\tau(u_0)=\sum_{i\in V}\lambda_i\langle u_0,\phi^i\rangle_{\mathcal{V}}^2+\limsup_{k\to\infty}\frac{1}{\tau_k}\left(\text{vol}\,V-\sum_{i\in V}\langle u_0,\phi^i\rangle_{\mathcal{V}}^2\right)$$
$$=2\text{TV}^+(u_0)=\hat{J}_0(u_0),$$

where we used Proposition 7 for the second equality, combined with the fact that, as in (25), $\sum_{i\in V}\langle u_0,\phi^i\rangle_{\mathcal{V}}^2=\langle u_0^2,\chi_V\rangle_{\mathcal{V}}=\langle\chi_V,\chi_V\rangle_{\mathcal{V}}=\text{Vol}\,V$. □

Lemma 6.3. *Let $G\in\mathcal{G}$ and let $\{\tau_k\}_{k\in\mathbb{N}}$ be a sequence of positive real numbers such that $\tau_k\downarrow 0$ as $k\to\infty$ and let $\{u_k\}_{k\in\mathbb{N}}$ be a sequence in $\mathcal{K}$. Assume that there exists a $C>0$ such that for all $k\in\mathbb{N}$ we have $\hat{J}_\tau(u_k)\leq C$. Then there exists a subsequence $\{u_{k'}\}_{k'\in\mathbb{N}}\subset\{u_k\}_{k\in\mathbb{N}}$ and a $u_0\in\mathcal{V}^b$ such that $u_{k'}\to u_0$ as $k'\to\infty$.*

Proof. This proof closely resembles the proof in [57, Theorem 5.23].

As explained in Section 4, to each $u\in\mathcal{V}$ we can assign a canonical vector representation $\hat{u}\in\mathbb{R}^n$. Let $\|u\|_2$ denote the Euclidean norm of $\hat{u}$. Since $\{u_k\}_{k\in\mathbb{N}}\subset\mathcal{K}$ we have that for all $k\in\mathbb{N}$, $0\leq\|u_k\|_2\leq\sqrt{n}$. By the Bolzano-Weierstrass theorem it follows that a subsequence $\{u_{k'}\}_{k'\in\mathbb{N}}\subset\{u_k\}_{k\in\mathbb{N}}$ exists which converges to a limit $u_0\in\mathcal{V}$ with respect to $\|\cdot\|_2$ and hence also with respect to the topologically equivalent norm $\|\cdot\|_{\mathcal{V}}$. Moreover, by componentwise convergence and preservation of inequalities in the limit, $\mathcal{K}$ is closed (with respect to either norm) and thus $u_0\in\mathcal{K}$.

Assume that $u_0\in\mathcal{K}\backslash\mathcal{V}^b$, then there exists a $j\in V$ such that $(u_0)_j\in(-1,1)$ and thus there exists an $\eta\in(0,1)$ such that, for all large enough $l'\in\mathbb{N}$, $(u_{l'})_j^2\in[0,1-\eta]$.

Since for all $i \in \{1, \ldots, n\}$ we have $\lambda_i \geq 0$, it follows by (25) that

$$
\begin{aligned}
\text{Vol } V - \sum_{i \in V} e^{-\tau_{l'} \lambda_i} \langle u_{l'}, \phi^i \rangle_{\mathcal{V}}^2 &\geq \text{Vol } V - \sum_{i \in V} \langle u_{l'}, \phi^i \rangle_{\mathcal{V}}^2 \\
&= \text{Vol } V - \langle u_{l'}^2, \chi_V \rangle_{\mathcal{V}}^2 = \langle \chi_V - u_{l'}^2, \chi_V \rangle_{\mathcal{V}} \\
&\geq \eta d_j^r.
\end{aligned}
\tag{26}
$$

Moreover, by assumption and using Proposition 5 we have, for all $k' \in \mathbb{N}$,

$$
\tau_{k'} \hat{J}_{\tau_{k'}}(u) = \text{Vol } V - \sum_{i \in V} e^{-\tau_{k'} \lambda_i} \langle u_{k'}, \phi^i \rangle_{\mathcal{V}}^2 \leq C \tau_{k'}.
$$

Choose l' to be large enough such that $C \tau_{l'} < \eta d_j^r$ and large enough so that (26) holds. Then we arrive at a contradiction and thus $u_0 \in \mathcal{V}^b$. $\qquad \square$

Theorem 6.4. *Let $G \in \mathcal{G}$. Let $\{\tau_k\}_{k \in \mathbb{N}} \subset (0, \infty)$ be a sequence such that $\tau_k \downarrow 0$ as $k \to \infty$ and, for each $k \in \mathbb{N}$, let $\hat{J}_{\tau_k}$ be as in (22) and let u_{τ_k} be a minimizer of $\hat{J}_{\tau_k}$. Then $\min_{u \in \mathcal{K}} \hat{J}_{\tau_k}(u) \to \min_{u \in \mathcal{K}} \hat{J}_0(u)$ as $k \to \infty$. Furthermore, there exists $u_0 \in \mathcal{V}^b$ and a subsequence $\{u_{\tau_{k'}}\}_{k' \in \mathbb{N}} \subset \{u_{\tau_k}\}_{k \in \mathbb{N}}$, such that $\|u_{\tau_{k'}} - u_0\|_{\mathcal{V}} \to 0$ as $k' \to \infty$. Moreover, $u_0 \in \operatorname*{argmin}_{u \in \mathcal{K}} \hat{J}_0(u)$, where $\hat{J}_0$ is as in (23). In particular, if $C_{u_0} \in \mathcal{C}$ is the cut induced by u_0, then C_{u_0} is a maximum cut of G.*

Proof. By standard Γ-convergence properties [20, Corollary 7.20], [13, Theorem 1.21], the equi-coercivity proven in Lemma 6.3 and the Γ-convergence of Lemma 6.2 together imply that $\min_{u \in \mathcal{K}} \hat{J}_{\tau_k}(u) \to \min_{u \in \mathcal{K}} \hat{J}_0(u)$ as $k \to \infty$. Furthermore, up to a subsequence, minimizers of $\hat{J}_{\tau_k}$ converge to a minimizer of $\hat{J}_0$ as $k \to \infty$. By Lemma 6.3, this minimizer, u_0 is in $\mathcal{V}^b$. The final statement follows in the same way as in the proof of Theorem 4.5. $\qquad \square$

Remark 3. Even though the results of Theorem 6.4 suggest that minimizers of $\hat{J}_\tau$ (or J_τ) for small $\tau > 0$ are good approximations to the solution of the Max-Cut problem, this is not a guarantee that solutions of the (MBO+) scheme are as well. In fact, as we have seen in Section 5.4, for small $\tau > 0$ the (MBO+) dynamics is trivial and thus will not produce an (approximate) solution to the Max-Cut problem unless the initial condition already is one.

7. Results.

7.1. **Method.** In Section 7 we compare the results of our new algorithm (MBO+) with the results obtained by (GW). In Sections 7.2–7.6 we display the results of (MBO+) using both the spectral decomposition method and the explicit Euler method, fixing the variable τ for both methods. We run all our tests on a Windows 7 PC with 16GB RAM and an Intel(R) Core(TM) i5-4590 CPU with clock speed 3.30GHz. For both (MBO+) and (GW) we use MATLAB, which is convenient to use when dealing with large sparse matrices.

For all of our tests using the spectral decomposition method we choose $K = \lfloor \frac{n}{100} \rfloor$. In practice it reduces the computation time without sacrificing much accuracy in the cut approximations. We further analyse this choice in Section 8. For all of our tests using the Euler method we set $M = 100$, in order to keep dt small so as to ensure stability on our explicit scheme. We compute the (MBO+) evolutions for 50 initial conditions chosen at random from $\mathcal{V}^b$. In the tables which we refer to in

(A) Web graph, maximum cut approximation

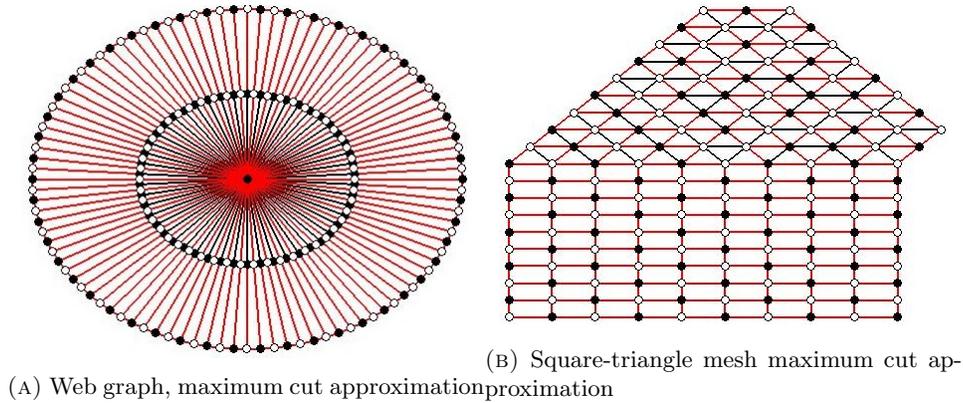(B) Square-triangle mesh maximum cut approximation

FIGURE 3. Visualisation of maximum cut approximations (best viewed in colour)

this section, we state the greatest (Best), average (Avg), and smallest (Least) sizes of cuts obtained by these 50 runs of (MBO+). We run (MBO+) using Δ_0^+, Δ_1^+ and Δ_s^+, fixing the initial conditions for each operator, using both the spectral method and the Euler method for each operator, and compare the results.

We compare the results of (MBO+) with those of (GW). To compute the relaxation step of (GW) we use SDPT3 MATLAB software [54] as it exploits the sparse structure of the matrices we work on. According to [43] it is best suited for both smaller problems and for larger problems with sparse matrices. The stopping tolerance is set as $|Z_P^* - Z_D^*| < 10^{-6}$. The recommended tolerance for the SDPT3 software is set as 10^{-8}. However, in our experiments increasing this tolerance to 10^{-6} reduced the computation time of (GW), without any change in output cut sizes. After the relaxation step, we perform the hyperplane step 50 times, randomly choosing a vector r each time. Each choice of r leads to a resulting cut; in the tables referred to in this section, we list the highest (Best), average (Avg), and lowest (Least) sizes of these cuts. In each of these categories in our tables we highlight the method that obtained the best result, (MBO+) using Δ_0^+, (MBO+) using Δ_1^+, (MBO+) using Δ_s^+, or (GW). We do the same for the run times (Time) of each method.

For both (MBO+) and (GW) only the adjacency matrix and the parameter choice η is initially provided, therefore the reported run times cover all calculations from that starting point. For each graph we remove the isolated nodes by removing all rows and columns of the graph's adjacency matrix which have all zero entries. (This does not affect the size of any cut of the graph.) For the spectral decompostion variant of (MBO+) using Δ_1^+ and Δ_s^+ this includes removing all isolated nodes, computing the matrices L_1 and L_s, finding their K eigenpairs corresponding to the leading eigenvalues in order to use Proposition 2, to compute the eigenpairs corresponding to the trailing eigenvalues of L_1^+ and L_s^+ respectively, generating initial conditions, running the signless diffusion and thresholding steps, and computing the size of the cut from each MBO iteration. For Δ_s^+ the computation time includes calculating L_1 in order to compute the size of the output cuts using (10). The computation time for (MBO+) using Δ_0^+ includes removing all isolated nodes,

computing the matrix L_0^+, finding all its eigenpairs, choosing the largest eigenvalue for the time step τ, and using the K eigenpairs corresponding to the smallest eigenvalues for the remaining steps.

For the explicit Euler method variant of (MBO+) the computation time includes removing all isolated nodes, computing $L^+ \in \{L_0^+, L_1^+, L_s^+\}$, generating initial conditions, running the signless diffusion and thresholding steps, and computing the size of the cut induced by each MBO iteration. For $L = L_s$ we also compute L_1 to obtain the size of the output cut using (10).

For every graph there exists a τ_{max} such that for all $\tau \geq \tau_{max}$ the solution to (16) computed using (MBO+) converges to $u(\tau) = 0$ to machine precision. In practice τ_{max} is dependent on the operator Δ^+. In our experiments we see that choosing a τ which is in between the pinning condition in Theorem 5.3 and τ_{max} is difficult due to the difference between them being small when Δ_0^+ is our operator for (MBO+). In Section 7.3 and Section 7.5 we run our experiments on graphs with a scale free structure (see Section 7.3). When running (MBO+) using the explicit Euler method and Δ_0^+ we encounter problems in choosing suitable τ and dt for such graphs. This is due to the inflexibility of choosing τ such that it is less than τ_{max} and also greater than the bound in Theorem 5.3. Since the Euler method is an approximation of the spectral method, we encounter problems in this case. If (MBO+) returns a cut which has pinned due to Theorem 5.3 or is zero due to the solution of (16) converging to zero to machine precision then we refer to the cut as a trivial cut. In Section 8.3 we show that it is possible to obtain non-trivial cut sizes using (MBO+) with Δ_0^+ by solving (16) using an implicit Euler scheme.

Figure 3 shows two examples of approximate maximum cuts obtained with the (MBO+) algorithm. The black nodes are in V_1 and the white nodes are in V_{-1}. An edge is coloured red, if it connects two nodes of different colour, i.e. if it contributes to the size of the cut. If it does not, it is black.

Figure 3a shows an unweighted web graph which has 201 nodes and 400 edges. We set $\tau = 20$ in (MBO+) using Δ_1^+ and the Euler method to solve (16). The resulting approximation of the maximum cut value is 350. The run time is 0.09 seconds. Figure 3b shows an unweighted triangle-square graph which has 162 nodes and 355 edges. We set $\tau = 20$ and $K = 20$ in (MBO+) using Δ_1^+ and the spectral method to solve (16). The approximation of the maximum cut value is 295 and the run time is 0.14 seconds.

7.2. **Random graphs.** In Figures 4, 5, and 6 we list results obtained for Erdös-Rényi graphs.

For each of $G(1000, 0.01)$ (Figure 4), $G(2500, 0.4)$ (Figure 5), and $G(5000, 0.001)$ (Figure 6) we create 100 realisations. We then run (MBO+) with both the spectral method and the Euler method, and we run (GW). For both of the (MBO+) methods we choose either Δ_0^+, Δ_1^+, or Δ_s^+, setting $\tau = 20$ for all tests. The bar chart represents the mean of the best, average, and least cuts over all 100 realisations of the chosen random graph. The error bars are the corrected sample standard deviation[5] of the results obtained over all 100 realisations. Figure 4 shows that (MBO+) using either the spectral method or Euler method for Δ_1^+ and Δ_s^+ produces better mean best, mean average, and mean least cuts than (GW) on this set of graphs. Figure 5 shows that (MBO+) using the spectral method and either Δ_1^+

---

[5]The corrected sample standard deviation is computed using MATLAB's `std` code in all experiments in this paper.
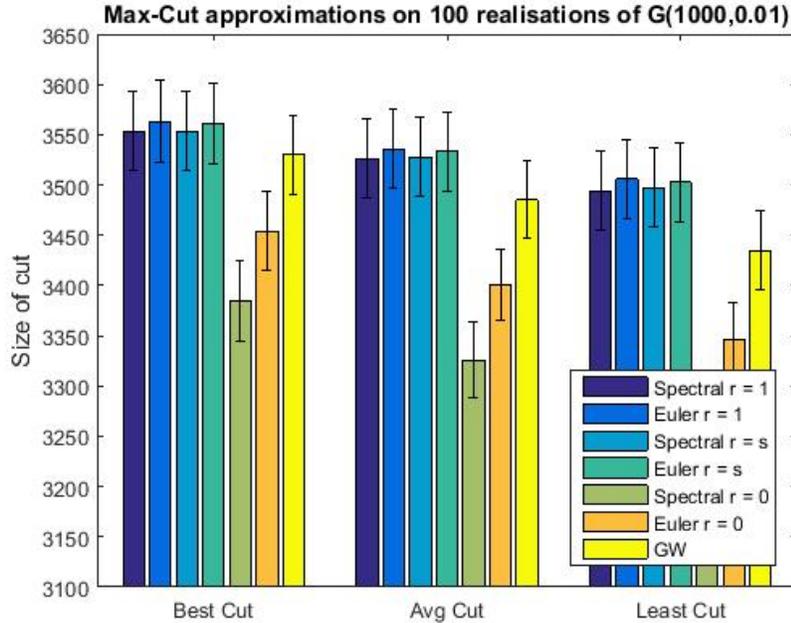
FIGURE 4. Bar chart of Max-Cut approximations on 100 realisations of $G(1000, 0.01)$.

Graph	Δ_1^+ (S)	Δ_1^+ (E)	Δ_s^+ (S)	Δ_s^+ (E)	Δ_0^+ (S)	Δ_0^+ (E)	GW
$G(1000, 0.01)$	**0.20**	1.58	0.34	1.52	0.56	1.06	5.25
$G(2500, 0.4)$	8.04	172.91	13.33	181.40	**6.40**	172.73	55.36
$G(5000, 0.001)$	**4.38**	16.96	6.37	14.95	24.99	6.97	257.09

TABLE 1. Average (MBO+) and (GW) run-times for each realisation of $G(n, p)$, time in seconds.

or Δ_s^+ produces better mean cut approximations than (GW) on this set of graphs. Figure 6 shows the same conclusions as Figure 4 for this set of graphs. Table 1 shows that (MBO+) using the spectral method produces the fastest run times on all three types of Erdös-Rényi graphs that we test on. We note that (GW) has a superior run time over (MBO+) using the Euler method on the realisations of $G(2500, 0.4)$.

7.3. **Scale-free graphs.** The degree distribution $P : \mathbb{N} \to \mathbb{R}$ of an unweighted graph G is given by $P(j) := \frac{|\{i \in V : d_i = j\}|}{n}$. Random graphs such as the ones discussed in Section 7.2 have a degree distribution which resembles a normal distribution. The graph $G \in \mathcal{G}$ is a scale-free graph if its degree distribution roughly follows a power law, i.e $P(j) \approx j^{-\gamma}$, where often in practice, $\gamma \in (2, 3)$ [7]. Scale-free graphs have become of interest as graphs such as internet networks, collaboration networks, and social networks are conjectured to more closely resemble scale-free graphs instead of random graphs [8].
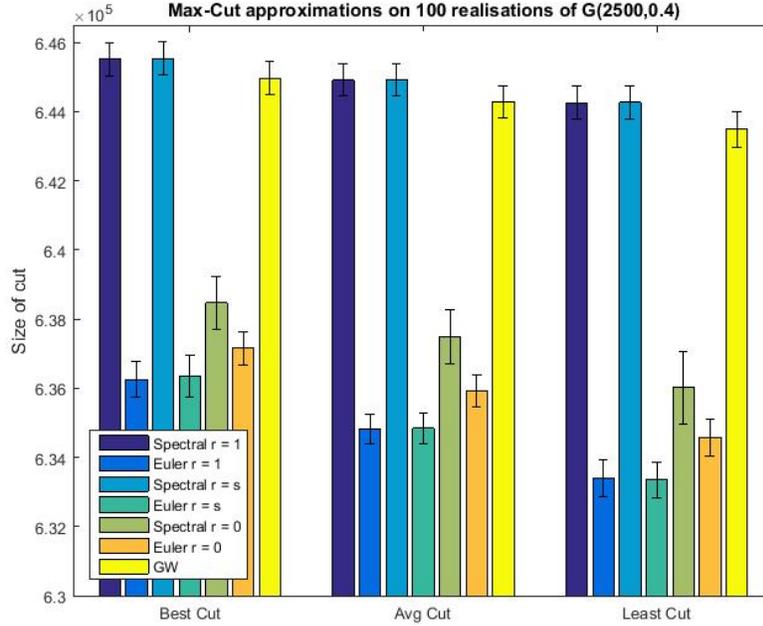
FIGURE 5. Bar chart of Max-Cut approximations on 100 realisations of $G(2500, 0.4)$.

In Table 6 we list results for some scale free graphs. We test the algorithms on 8 autonomous systems internet graphs, ASi, $i \in \{1, \ldots, 8\}$. These graphs represent smaller imitations of an internet network, which were acquired from the website [1]. We also test on the graph Gnutella09 which is a model of a peer to peer file sharing network, and the graph WikiVote, which is a network representing a Wikipedia administrator election, both obtained from [3]. All of the scale free graphs in this section are unweighted and undirected graphs.

Table 2 displays some properties of the random graphs in Section 7.2 and the scale-free graphs we test on. Figure 7 displays the average degree distribution of 100 realisations of $G(2500, 0.4)$, in Figure 7a, and the degree distribution of the AS1 Graph, in Figure 7b. In Figure 7a the yellow points indicate the degree distribution, and the orange lines indicate the corrected sample standard deviation of the average degree distribution. In Figure 7b the blue dots indicate the degree distribution. As we see, the average degree distribution of the realisations of $G(2500, 0.4)$ is similar to a normal distribution, and the degree distribution of the AS1 graph resembles a power law, as expected.

For all graphs listed in Table 6, using either Δ_1^+ or Δ_s^+ (MBO+) using the Euler method or the spectral method outperforms (GW) with respect to the average and least obtained cut sizes and the run time, but (GW) obtains the best results when considering the greatest obtained cuts. For any choice of Δ_1^+ and Δ_s^+ and for any choice of signless diffusion solver the greatest cuts obtained by (MBO+) are all at least 98.1% of the greatest cut size obtained by (GW). The difference in run times is notable though. The time taken by (MBO+) stays below 30 seconds for all graphs
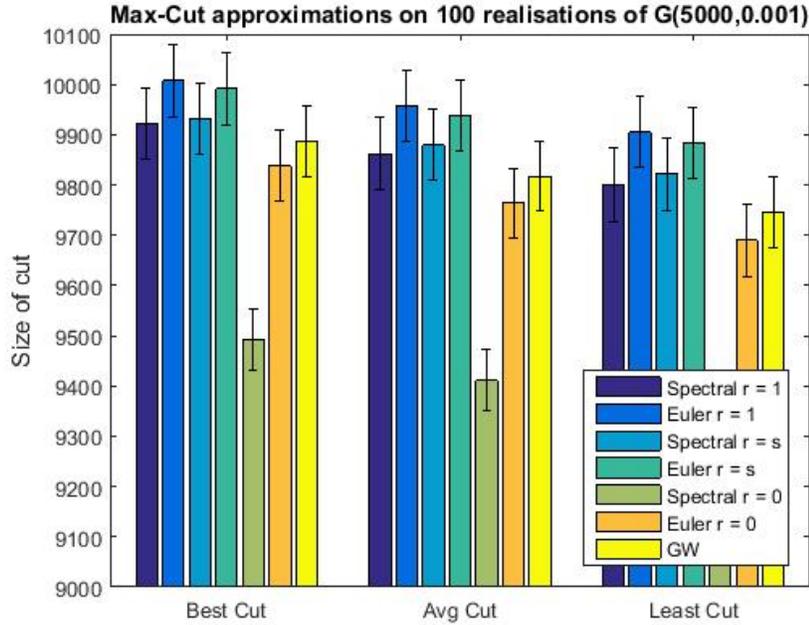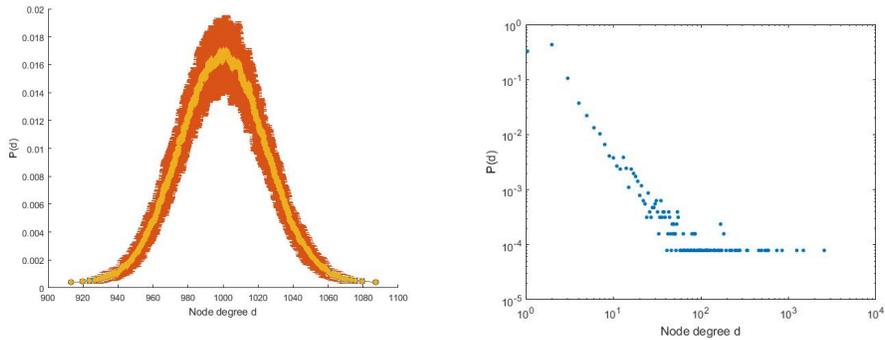
FIGURE 6. Bar chart of Max-Cut approximations on 100 realisations of $G(5000, 0.001)$.

(A) Degree distribution of a realisation of $G(2500, 0.4)$.

(B) Degree distribution of the AS1 Graph.

FIGURE 7. Average degree distribution of 100 realisations of a random graph and the degree distribution of a scale free graph.

in Table 6, irrespective of choice of Laplacian and signless diffusion solver. However, the (GW) algorithm's run times range between 9 and 44 minutes. These results suggest that (MBO+) using Δ_1^+ or Δ_s^+, and using either signless diffusion solver offers a significant decrease in run time at the cost of about 1-2% accuracy in the resulting cut size, in comparison with (GW), when applied to the graphs in Table 6.

Graph	$|V|$	$|E|$	d_-	d_+
$G(1000, 0.01)(1)$	1000	4919	1	21
$G(1000, 0.01)(2)$	1000	4939	2	21
$G(2500, 0.4)(1)$	2500	1248937	910	1079
$G(2500, 0.4)(2)$	2500	1251182	904	1081
$G(5000, 0.001)(1)$	4962	12646	1	16
$G(5000, 0.001)(2)$	4969	12642	1	16

Graph	$|V|$	$|E|$	d_-	d_+
AS1	12694	26559	1	2566
AS2	7690	15413	1	1713
AS3	8689	17709	1	1911
AS4	8904	17653	1	1921
GNutella09	8114	26013	1	102
Wiki-Vote	7115	100762	1	1065

TABLE 2. Properties of $G(n, p)$ graph realisations vs scale free graphs.

Graph	Δ_1^+ (S) Best	Δ_1^+ (S) Avg	Δ_1^+ (S) Least	Δ_1^+ (S) Time
AS1	22744	22542.20	22183	**15.85**
AS2	13249	13153.72	13054	**3.55**
AS3	15118	15027.22	14907	4.73
AS4	15194	15143.44	15042	**5.67**
AS5	14080	13988.90	13928	**4.82**
AS6	18053	17964.74	17876	10.06
AS7	22741	22535.00	22150	17.82
AS8	22990	22720.36	22334	17.22
GNutella09	20280	20143.74	19983	8.16
WikiVote	72981	72856.40	72744	2.46

Graph	Δ_1^+ (E) Best	Δ_1^+ (E) Avg	Δ_1^+ (E) Least	Δ_1^+ (E) Time
AS1	22798	**22670.76**	22268	23.62
AS2	13281	**13199.72**	**13120**	8.76
AS3	15175	**15095.46**	**15007**	9.95
AS4	15270	**15202.70**	**15117**	10.88
AS5	14120	**14020.62**	**13944**	9.50
AS6	18134	**18034.10**	**17933**	16.50
AS7	22826	**22696.42**	**22525**	25.78
AS8	23070	**22951.54**	**22550**	25.38
GNutella09	20437	**20361.92**	**20295**	17.14
WikiVote	73159	**73126.34**	**73086**	9.06

TABLE 3. (MBO+) cut approximations using Δ_1^+ on graphs with a scale free structure, time in seconds.

Graph	Δ_s^+ (S) Best	Δ_s^+ (S) Avg	Δ_s^+ (S) Least	Δ_s^+ (S) Time
AS1	22809	22620.8	**22325**	17.83
AS2	13271	13178.86	13103	4.12
AS3	15166	15082.1	14992	**4.66**
AS4	15237	15166.24	15077	5.78
AS5	14075	14011.96	13911	5.47
AS6	18088	17968.04	17859	**9.14**
AS7	22822	22629.66	22218	**15.73**
AS8	23061	22884.8	22547	**15.46**
GNutella09	20282	20186.32	20101	**6.82**
WikiVote	73169	73003.44	72917	**2.25**

Graph	Δ_s^+ (E) Best	Δ_s^+ (E) Avg	Δ_s^+ (E) Least	Δ_s^+ (E) Time
AS1	22789	22629.62	22261	27.63
AS2	13256	13176.64	13094	9.09
AS3	15139	15059.54	14967	10.24
AS4	15234	15159.76	15079	11.57
AS5	14096	14011.9	13930	10.47
AS6	18088	17994.66	17876	16.12
AS7	22823	22639.58	22237	24.5
AS8	23036	22865	22440	25.08
GNutella09	20397	20332.28	20170	18.75
WikiVote	72993	72772.26	72549	9.00

TABLE 4. (MBO+) cut approximations using Δ_s^+ on graphs with a scale free structure, time in seconds.

Graph	Δ_0^+ (S) Best	Δ_0^+ (S) Avg	Δ_0^+ (S) Least	Δ_0^+ (S) Time
AS1	22578	22303.10	21844	297.79
AS2	13081	12935.80	12763	62.41
AS3	14995	14869.52	14702	90.32
AS4	15097	14994.92	14885	88.53
AS5	13952	13795.24	13561	70.81
AS6	17836	17672.50	17527	149.60
AS7	22571	22328.18	21932	294.26
AS8	22824	22585.88	22075	287.79
GNutella09	19079	18419.36	17951	72.03
WikiVote	65504	60599.74	56917	46.11

TABLE 5. (MBO+) cut approximations using Δ_0^+ on graphs with a scale free structure, time in seconds.

7.4. **Random modular graphs.** Modular graphs have a community structure. Nodes in a community have many connections with other members of the same community and noticeably fewer connections with members of other communities. In Figure 8 we show what our Max-Cut approximation looks like on a random modular graph. We generate realisations of random unweighted modular graphs

Graph	GW Best	GW Avg	GW Least	GW Time
AS1	**22864**	22346.26	20546	2324.98
AS2	**13328**	13039.10	12048	594.29
AS3	**15240**	14961.56	14050	826.65
AS4	**15328**	15015.34	14072	832.28
AS5	**14190**	13810.82	12922	721.51
AS6	**18191**	17851.24	16483	1368.35
AS7	**22901**	22421.80	21244	2321.34
AS8	**23170**	22593.10	21110	2613.62
GNutella09	**20658**	20242.02	18815	1095.04
Wiki-Vote	**73363**	71510	62886	1074.98

TABLE 6. (GW) cut approximations on graphs with a scale free structure, time in seconds.
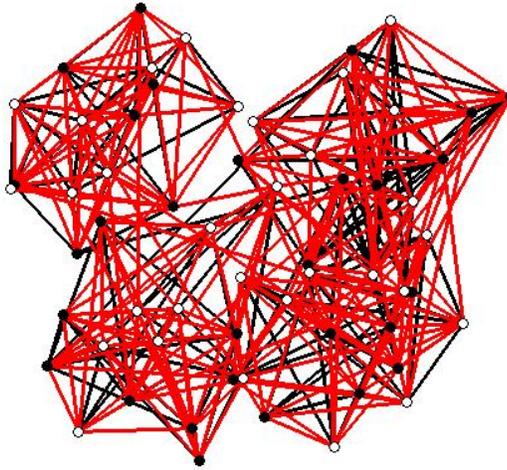

FIGURE 8. A Max-Cut approximation on a random 4-modular graph (best viewed in colour).

$R(n, c, p, r)$ using the code provided at [2]. The variables for the graph are the number of nodes n, the number $c \in \mathbb{N}$ of communities that the graph contains, a probability p such that the graph will have an expected number of $\frac{n^2}{2p}$ edges, and a ratio $r \in [0, 1]$, with $r|E|$ being the expected number of edges connecting nodes in the same community and $(1 - r)|E|$ being the expected number of edges connecting nodes in different communities.

In Figures 9, 10, and 11 we display results obtained for random modular graphs. For each of $R(2500, 2, 0.009, 0.8)$ (Figure 9), $R(4000, 20, 0.01, 0.7)$ (Figure 10), and $R(10000, 10, 0.01, 0.8)$ (Figure 11) we create 100 realisations. We then run (MBO+)
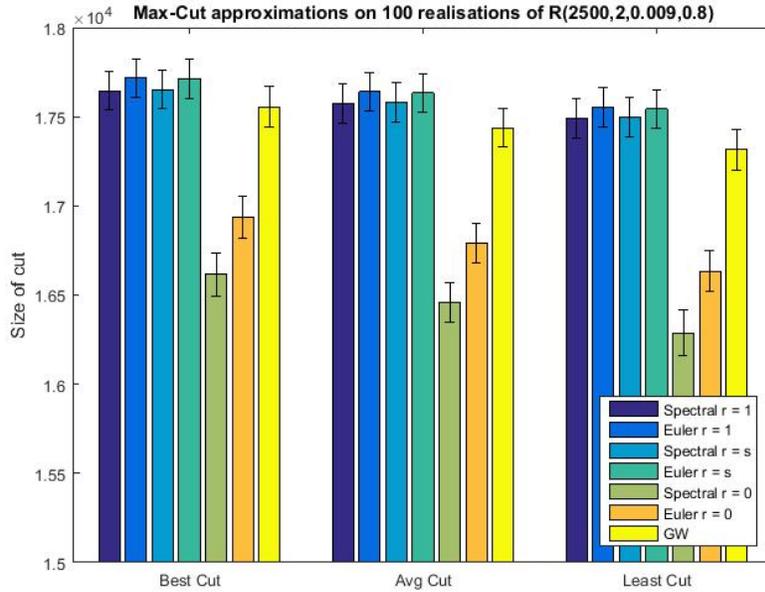
FIGURE 9. Bar chart of Max-Cut approximations on 100 realisations of $R(2500, 2, 0.009, 0.8)$.

Graph	Δ_1^+ (S)	Δ_1^+ (E)	Δ_s^+ (S)	Δ_s^+ (E)	Δ_0^+ (S)	Δ_0^+ (E)	GW
$R(2500, 2, 0.009, 0.8)$	0.80	10.43	**0.79**	10.26	4.36	6.13	56.30
$R(4000, 20, 0.01, 0.7)$	**4.05**	30.46	4.49	29.52	16.26	18.19	248.25
$R(10000, 10, 0.01, 0.8)$	**49.98**	266.10	52.85	266.40	210.94	194.52	3893.87

TABLE 7. Average (MBO+) and (GW) run-times for each realisation of $R(n, c, p, r)$, time in seconds.

with both the spectral method and the Euler method, and we run (GW). For both of the (MBO+) methods we choose either Δ_0^+, Δ_1^+, or Δ_s^+, setting $\tau = 20$ for all tests. The bar chart represents the mean of the best, average, and least cuts over all 100 realisations of the chosen random modular graph. The error bars are the corrected sample standard deviation of the results obtained over all 100 realisations.

In Figures 9, 10, and 11 we see that using either Δ_1^+ or Δ_s^+ (MBO+) with both the spectral method and the Euler method outperforms (GW) with respect to the best, average, and least cuts. In Table 7 we see that for any choice of operator and method, (MBO+) is faster on average than (GW) for our choices for random modular graphs. We note in particular that for our realisations of $R(10000, 10, 0.01, 0.8)$ the average (GW) test took just below 65 minutes, where as the average (MBO+) test using the spectral method and either Δ_1^+ or Δ_s^+ took under a minute, obtaining on average better outcomes.

7.5. **Weighted graphs.** In this subsection we assign random weights to the edges of selected graphs from Section 7.2 and Section 7.3. To create the graphs W1 and W2 we use two of the realisations of $G(1000, 0.01)$, and multiply its edges by random
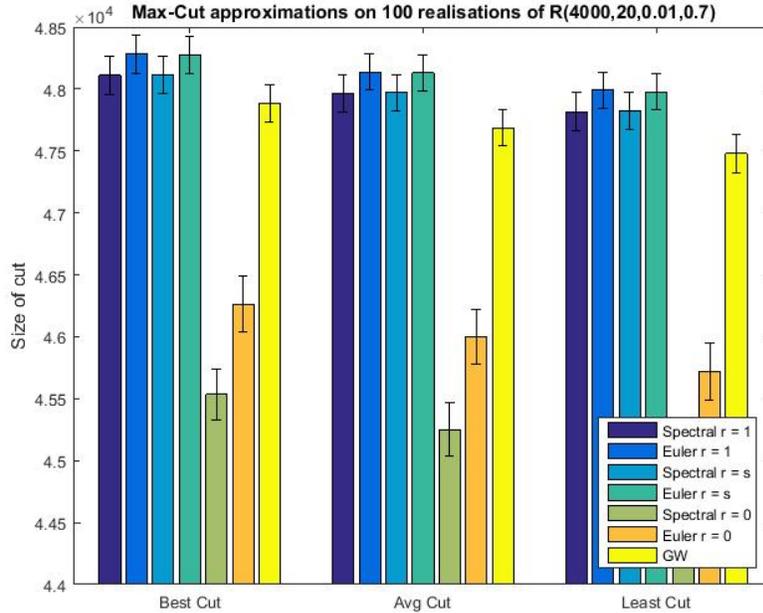
FIGURE 10. Bar chart of Max-Cut approximations on 100 realisations of $R(4000, 20, 0.01, 0.7)$.

real numbers drawn uniformly from in the range $[0, 2]$ and $[0, 20]$ respectively. W3 and W4 were created by using two of the realisations of $G(2500, 0.4)$ in Section 7.2, and multiplying its edges by random real numbers drawn uniformly from in the ranges $[0, 5]$ and $[0, 1]$ respectively. W5, W6, W7 were created by using three of the realisations of $G(5000, 0.001)$ in Section 7.2, and multiplying its edges by random real numbers drawn uniformly from in the ranges $[0, 1], [0, 15]$, and $[0, 50]$ respectively. W8 is the AS1 graph, whose edges are multiplied by random real numbers drawn uniformly from in the range $[0, 12]$, W9 is the AS5 graph whose edges are multiplied by random real numbers drawn uniformly from in the range $[0, 4]$ and W10 is the AS8 graph whose edges are multiplied by random real numbers drawn uniformly from in the range $[0, 8]$. We run (MBO+) for all three choices of Δ^+, on all of these graphs, and compare against (GW) in Table 11. We set $\tau = 20$ for both the spectral decomposition method and the Euler method.

We saw that (MBO+) using the spectral method produced larger cuts than (GW) on the random graphs considered in Section 7.2; when assigning random weights to the edges of these random graphs the same conclusion holds. We see in Table 11 that for this collection of random graphs (MBO+) using the spectral method (with either Δ_1^+ or Δ_s^+ used) outperforms (GW) with respect to the best, average, and smallest obtained cut sizes, and the run time. In Section 7.3 we saw that (MBO+) using both the spectral method and the Euler method produced better average and smallest cuts than (GW) on the scale free graphs considered in that section, but the best cut sizes were produced more often by (GW). These weighted examples support the same conclusions. The blank results in Table 11 for the Euler method
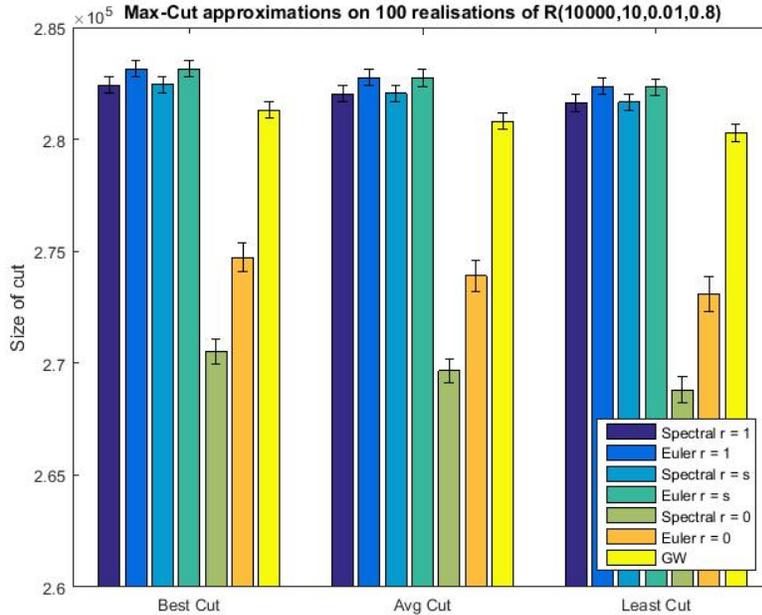
FIGURE 11. Bar chart of Max-Cut approximations on 100 realisations of $R(10000, 10, 0.01, 0.8)$.

using Δ_0^+ are due to (MBO+) producing trivial results for these choices as stated in Section 7.1.

7.6. **Large graphs.** Since the Euler method has a time complexity of $\mathcal{O}(|E|)$, in this section we show that (MBO+) using the Euler method can provide Max-Cut approximations in a respectable time on large sparse datasets. The graphs Amazon0302 and Amazon0601 are networks in which the nodes represent products and an edge exists between two nodes if the corresponding products are frequently co-purchased; both of these networks were constructed in 2003. GNutella31 depicts a peer to peer file sharing network in 2002. PA RoadNet is a road network of Pennsylvania with intersections and endpoints acting as nodes and roads connecting them acting as edges. Email-Enron is a network where each edge represents an email being sent between two people. BerkStan-Web is a network of inter-domain and intra-domain hyperlinks between pages on the domains berkeley.edu and stanford.edu in 2002. Stanford is a network of hyperlinks between pages on the domain stanford.edu in 2002. All of these datasets were obtained from the website [3]. The graph WWW1999 is a model of the Internet in 1999 with edges depicting hyperlinks between websites, obtained from [4]. Table 12 displays the properties of these graphs. Table 13 displays the results we obtained on these graphs choosing Δ_1^+ as our operator, the Euler method as our signless diffusion solver, and $\tau = 10$. For these large graphs, we are unable to obtain results for comparison using (GW), because (GW) requires too much memory for it to run on the same computer setup.

Graph	Δ_1^+ (S) Best	Δ_1^+ (S) Avg	Δ_1^+ (S) Least	Δ_1^+ (S) Time
W1	3612.00	3569.08	3537.10	0.47
W2	36487.51	36082.58	35687.87	**0.30**
W3	1622125.53	**1620885.77**	**1619371.25**	8.09
W4	323926.34	323639.05	**323321.92**	8.59
W5	5054.26	5033.54	5010.38	**4.00**
W6	74560.24	74218.26	73776.17	**3.90**
W7	252448.52	251045.03	249459.89	4.18
W8	137202.14	135952.94	133480.08	16.17
W9	28351.01	28194.96	28009.15	**3.99**
W10	92376.49	91570.35	90172.90	17.02

| Graph	Δ_1^+ (E) Best	Δ_1^+ (E) Avg	Δ_1^+ (E) Least	Δ_1^+ (E) Time
W1	**3622.58**	**3580.53**	3548.82	1.41
W2	**36530.25**	**36191.16**	**35928.56**	1.67
W3	1603390.76	1600505.43	1596558.94	185.03
W4	320347.01	319612.93	318849.26	195.66
W5	**5104.45**	**5081.95**	**5063.64**	15.31
W6	**75499.50**	**75175.73**	**74833.80**	15.70
W7	**255793.23**	**254569.97**	**253091.91**	15.71
W8	137569.32	**136896.1**	**136094.60**	23.83
W9	28545.45	**28369.43**	**28141.76**	9.24
W10	93021.06	**92489.04**	**91626.99**	25.37

TABLE 8. (MBO+) cut approximations using Δ_1^+ on randomly weighted graphs, time in seconds.

8. Parameter choices.

8.1. **Variable** K. As stated in Section 5.2, the computational advantage of (MBO+) using the spectral method is that not all the eigenpairs of Δ^+ need to be used. In practice, if K is large enough, the cut sizes obtained by (MBO+) using the spectral method does not improve significantly when K is increased further. The plots in Figure 12 highlight this. For these three tests we fixed the initial conditions, the choice of operator Δ_1^+, and $\tau = 20$ for each respective graph. For Figure 12a we plot the best, average, and least cuts for each choice of K. For Figure 12b and Figure 12c we plot the mean of the best, average, and least cuts over all 100 graphs for each choice of K. The error bars indicate the corrected sample standard deviation of the best, average, and least cuts. We ran (MBO+) using the spectral method on the AS4 graph, increasing the value for K in increments of 5 from 5 until 100. The plot in Figure 12a shows that at $K = 40$ the best, average, and least cut size changes very little for increasing K. For Figure 12b we ran (MBO+) on the 100 realisations of $G(5000, 0.001)$ from Section 7.2, increasing K in increments of 10 from 10 until 200. For Figure 12c we ran (MBO+) on the 100 realisations of $R(2500, 2, 0.009, 0.8)$, increasing K in increments of 5 from 5 until 100. The plots in Figure 12b and Figure 12c show that for our choices of Erdös-Rényi and random modular graphs increasing K increases the cut sizes. We also note that the best, average, and minimum cut sizes plateau.
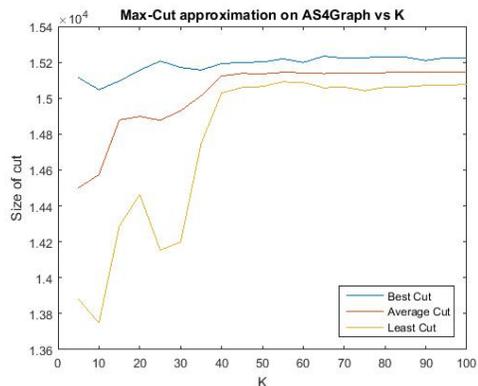
(A) AS4: Cut size as function of K, $\tau = 20$.
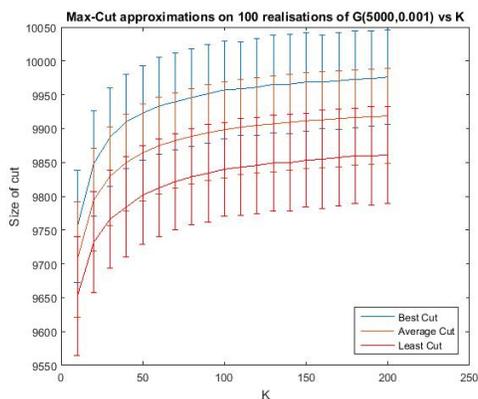
(B) 100 realisations of $G(5000, 0.001)$: Cut size as function of K, $\tau = 20$.
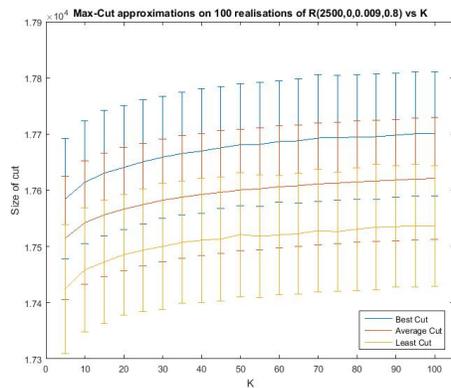
(C) 100 realisations of $R(2500, 2, 0.009, 0.8)$: Cut size as function of K, $\tau = 20$.

FIGURE 12. Cut size as function of K for three graphs (best viewed in colour).

Graph	Δ_s^+ (S) Best	Δ_s^+ (S) Avg	Δ_s^+ (S) Least	Δ_s^+ (S) Time
W1	3601.29	3569.23	3545.85	**0.33**
W2	36192.09	36059.80	35867.83	0.49
W3	**1622372.91**	1620484	1618809.76	8.40
W4	**323933.40**	**323642.4**	323114.45	7.65
W5	5068.19	5041.94	5015.16	4.50
W6	74844.37	74505.45	73963.79	4.67
W7	253043.96	251668.30	250600.35	**4.12**
W8	137195.52	136360.17	134856.06	**15.38**
W9	28389.38	28227.09	28067.66	4.12
W10	92439.42	91952.98	90488.33	**15.33**

Graph	Δ_s^+ (E) Best	Δ_s^+ (E) Avg	Δ_s^+ (E) Least	Δ_s^+ (E) Time
W1	3614.37	3577.56	3542.19	1.40
W2	36321.80	36150.05	35910.90	1.53
W3	1604257.12	1600145.68	1597577.4	187.88
W4	320691.88	319596.27	318900.13	199.01
W5	5096.55	5072.36	5041.89	15.9
W6	75456.87	75089.73	74745.17	18.09
W7	255316.85	253821.64	252527.13	15.48
W8	137282.02	136475.24	134333.1	24.51
W9	28445.94	28258.64	28101.22	9.18
W10	92731.62	92093.05	90448.61	24.36

TABLE 9. (MBO+) cut approximations using Δ_s^+ on randomly weighted graphs, time in seconds.

For large graphs, however, finding the value of K beyond which the produced cut sizes plateau is problematic. We ran (MBO+) using the spectral method with Δ_1^+ on the Amazon0302 graph, increasing K in increments of 100 starting from 100 to 2600. As shown in Figure 13 the best, average, and least outcomes of (MBO+) are still increasing at the end of the range of K values we plotted. For $K = 200$ and $K = 2600$ the run time of (MBO+) was 12 minutes and 26 hours, respectively; this increase in computation time resulted in a 3% increase in cut values. Comparing the cut size obtained for $K = 2600$ with the cut sizes obtained on Amazon0302 in Table 13 we see that using the Euler method as the signless diffusion solver is more accurate and significantly faster.

8.2. **Variable τ.** Other than the pinning condition stated in Section 5.4, currently we have very little information on which to base our choice of τ. In this section we compare the cut sizes obtained by (MBO+) against the variable τ. We choose Δ_1^+ as the signless Laplacian operator and the spectral method as the signless diffusion solver. Figure 14 displays the obtained cut sizes from (MBO+) on three (sets of) graphs and compares against τ. For Figure 14a we plot the best, average, and least cuts for each choice of τ. For Figure 14b and Figure 14c we plot the mean of the best, average, and least cuts over all 100 graphs for each choice of τ. The error bars indicate the corrected sample standard deviation of the best, average, and least cuts. We ran (MBO+) using the spectral method on the AS4 graph, increasing the

Graph	Δ_0^+ (S) Best	Δ_0^+ (S) Avg	Δ_0^+ (S) Least	Δ_0^+ (S) Time
W1	3413.96	3345.32	3276.63	0.61
W2	34784.30	34304.33	33627.16	0.51
W3	1602346.52	1600022.33	1595791.12	**6.97**
W4	320251.52	319940.38	319663.40	**6.25**
W5	4793.44	4761.72	4715.51	18.66
W6	71219.49	70427.83	69643.31	18.93
W7	239991.72	237647.45	235617.15	19.17
W8	134097.55	131088.97	126123.70	272.56
W9	27528.99	26554.77	25501.34	69.63
W10	90271.70	88031.84	83130.60	264.89

Graph	Δ_0^+ (E) Best	Δ_0^+ (E) Avg	Δ_0^+ (E) Least	Δ_0^+ (E) Time
W1	3524.24	3456.55	3406.93	1.03
W2	35664.18	35040.71	34383.57	1.03
W3	1605419.97	1602251.82	1597064.59	203.27
W4	320321.63	319809.73	319237.08	192.51
W5	5017.66	4983.90	4954.63	7.76
W6	74195.87	73688.97	73231.67	7.33
W7	251330.73	249754.88	248091.06	7.51
W8	-	-	-	-
W9	-	-	-	-
W10	-	-	-	-

TABLE 10. (MBO+) cut approximations using Δ_0^+ on randomly weighted graphs, time in seconds.

| Graph	GW Best	GW Avg	GW Least	GW Time
W1	3585.17	3535.63	3494.26	5.74
W2	36101.30	35698.47	35151.60	6.07
W3	1620705.80	1618813.52	1616502.33	43.58
W4	323573.40	323275.84	322795.83	44.09
W5	5038.00	5000.74	4953.71	265.27
W6	74372.75	73852.36	73293.27	241.33
W7	251802.56	250316.08	248098.85	263.44
W8	**138159.14**	135899.20	129576.95	2629.60
W9	**28705.35**	28169.25	26422.54	689.16
W10	**93547.26**	91571.68	87487.99	2646.94

TABLE 11. (GW) cut approximations on randomly weighted graphs, time in seconds.

value for τ in increments of 5 starting from 5 until 500. In Figure 14a we see in this experiment that $5 \leq \tau \leq 40$ produces the best results with respect to our cut sizes. We also see that for $330 \leq \tau \leq 480$ the best, average, and least cuts are almost identical. For Figure 14b we ran (MBO+) on the 100 realisations of $G(5000, 0.001)$

Graph	$|V|$	$|E|$	d_-	d_+
Amazon0302	262111	899792	1	420
Amazon0601	403394	2443408	1	2752
GNutella31	62586	147892	1	95
PA RoadNet	1088092	1541898	1	9
Email-Enron	36692	183831	1	1383
BerkStan-Web	685230	6649470	1	84290
Stanford	281904	1992636	1	38625
WWW1999	325729	1090108	1	10721

TABLE 12. Properties of our large datasets we are testing on.

Graph	Δ_1^+ (E) Best	Δ_1^+ (E) Avg	Δ_1^+ (E) Min	Δ_1^+ (E) Time
Amazon0302	618942	618512.18	618030	0.49
Amazon0601	1580070	1576960.80	1571089	1.90
GNutella31	116552	116213.74	115916	0.06
PA RoadNet	1380131	1379797.90	1379416	0.64
Email-Enron	112665	111680.24	110279	0.02
BerkStan-Web	5335813	5319662.06	5281630	0.83
Stanford	1585802	1580445.14	1570469	0.47
WWW1999	813000	809329.52	806130	0.21

TABLE 13. Results of (MBO+) using Δ_1^+ and the Euler method
on large datasets, time in hours.
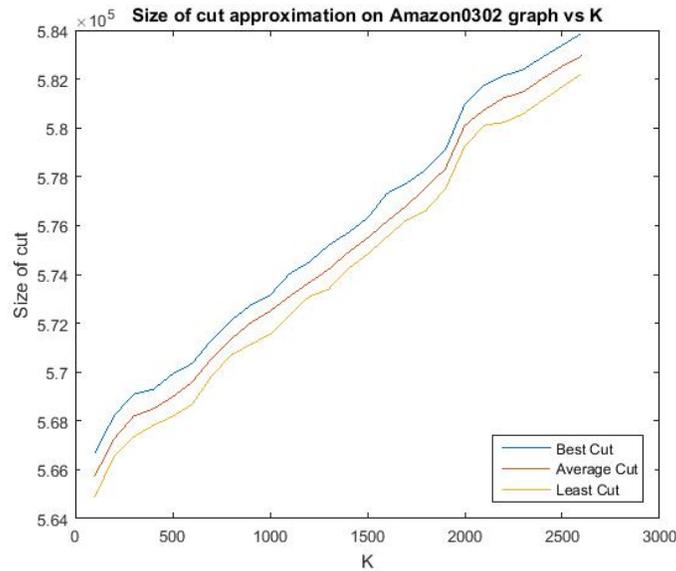
FIGURE 13. Comparison of cut size approximation vs K on Amazon0302 graph.

Graph	Δ_1^+ Best	Δ_s^+ Best	Δ_0^+ Best
AS4	15276	**15279**	9259
AS8	**23083**	23033	13725
W9	**28553.66**	28485.28	17146.69
Graph	Δ_1^+ Avg	Δ_s^+ Avg	Δ_0^+ Avg
AS4	**15196.52**	15175.52	9124.68
AS8	**22934.30**	22844.16	13585.56
W9	**28360.46**	28294.40	16847.92

Graph	Δ_1^+ Least	Δ_s^+ Least	Δ_0^+ Least
AS4	**15124**	15056	8964
AS8	**22521**	22454	13477
W9	**28103.28**	28075.62	16521.43
Graph	Δ_1^+ Time	Δ_s^+ Time	Δ_0^+ Time
AS4	47.83	50.94	**7.47**
AS8	105.22	114.74	**11.48**
W9	38.61	42.57	**5.26**

TABLE 14. Cut sizes obtained by (MBO+) using the implicit Euler scheme on scale free graphs, time in seconds.

from Section 7.2, increasing τ in increments of 5 starting from 5 until 125. For Figure 14c we ran (MBO+) on the 100 realisations of $R(4000, 20, 0.01, 0.7)$, increasing τ in increments of 5 starting from 5 until 100. In Figure 14b and Figure 14c we see the general trend that increasing τ beyond 20 decreases the mean over the best, average, and least cuts over all 100 realisations of $G(5000, 0.001)$.

8.3. **Implicit Euler scheme.** On the random graphs we tested on in Section 7.2 and Section 7.4 our explicit Euler scheme using Δ_0^+ produced non-trivial cut sizes. However, for the scale free graphs in Section 7.3 and Section 7.5 we did not find a value of τ or dt such that the cuts induced from (MBO+) were non-trivial. In this subsection we show that we can solve the Euler equation implicitly in order to obtain non-trivial cut sizes with the operator Δ_0^+, subject to suitable choices of dt and τ. However, the results are significantly inferior to the operators Δ_1^+ and Δ_s^+ for the implicit Euler scheme. We also compare the (MBO+) results obtained using the implicit scheme to the results obtained using the explicit scheme for a set of random graphs.

We run (MBO+) using the implicit Euler scheme on the AS4 and AS8 graph from Section 7.3 and the W9 graph from Section 7.5. We choose $dt = 0.2$ and $\tau = 20$ when Δ_1^+ or Δ_s^+ is the operator. For Δ_0^+ we set $dt = 0.0005$ and $\tau = 0.05$ for the AS4 graph, and for the AS8 graph and the W9 graph we set $dt = 0.0001$ and $\tau = 0.01$. Table 14 shows that (MBO+) using the implicit Euler scheme with Δ_0^+ and our choice of parameters produces cut sizes, however they are significantly smaller in comparison to using Δ_1^+ or Δ_s^+.

We run (MBO+) on the 100 realisations of $G(1000, 0.01)$ and $R(4000, 20, 0.01, 0.7)$ in Section 7.2 and Section 7.4 respectively, using the implicit and explicit Euler method for each operator $\Delta^+ \in \{\Delta_0^+, \Delta_1^+, \Delta_s^+\}$. We choose the same values of τ and dt as chosen in Section 7.2 and Section 7.4, fixing the initial conditions for both
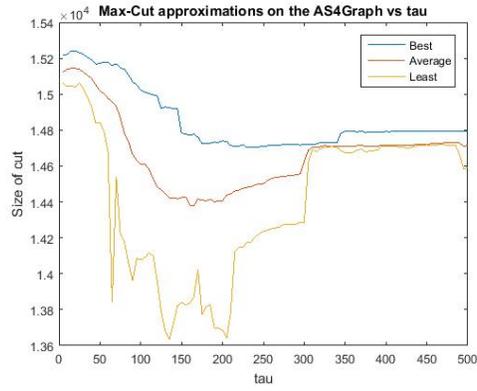
(A) AS4: Cut size as function of τ, $K = 89$.
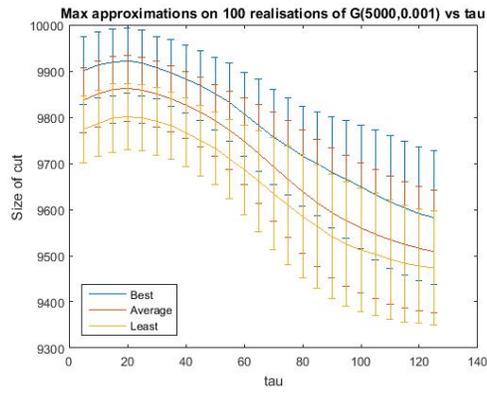
(B) 100 realisations of $G(5000, 0.001)$: Cut size as function of τ, $K = 49$.
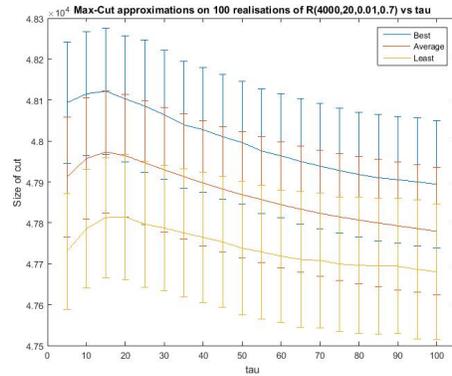
(C) 100 realisations of $R(4000, 20, 0.01, 0.7)$: Cut size as function of τ, $K = 40$.

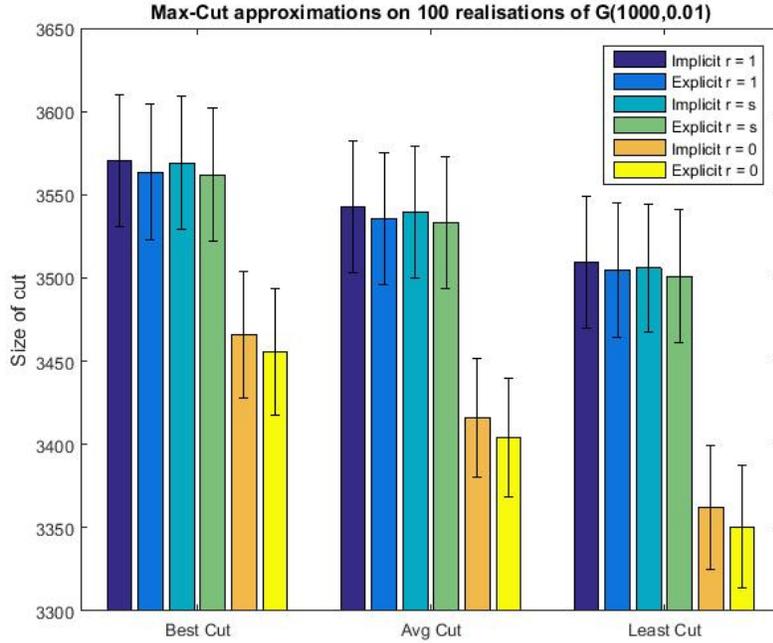FIGURE 14. Cut size as function of τ for three graphs (best viewed in colour).

FIGURE 15. Bar chart of Max-Cut approximations on 100 realisations of $G(1000, 0.01)$ using the implicit Euler method and the explicit Euler method.

Graph	Δ_1^+ (I)	Δ_1^+ (E)	Δ_s^+ (I)	Δ_s^+ (E)	Δ_0^+ (I)	Δ_0^+ (E)
$G(1000, 0.01)$	3.36	1.82	3.28	1.79	2.20	**1.19**
$R(4000, 20, 0.01, 0.7)$	62.97	44.23	62.16	44.18	41.53	**24.01**

TABLE 15. Average (MBO+) run-times for each realisation of $G(1000, 0.01)$ and $R(4000, 20, 0.01, 0.7)$, time in seconds. (I) indicates the implicit Euler method and (E) indicates the explicit Euler method.

methods. Figure 15 and Figure 16 show that the average obtained cut sizes using the implicit Euler method are slightly better than the average obtained cut sizes obtained using the explicit method. However, Table 15 shows that (MBO+) using the explicit Euler method produces cut sizes in less time than using the implicit Euler method on these sets of random graphs. This is why we choose the explicit method for the Euler method in Section 7.

9. **Conclusions.** We have proven that the signless graph Ginzburg–Landau functional f_ε^+ Γ-converges to a Max-Cut objective functional as $\varepsilon \downarrow 0$ and thus minimizers of f_ε^+ can be used to approximate maximal cuts of a graph. We use an adaptation of the graph MBO scheme involving signless graph Laplacians to approximately minimize f_ε^+. We solve the signless diffusion step of our graph MBO
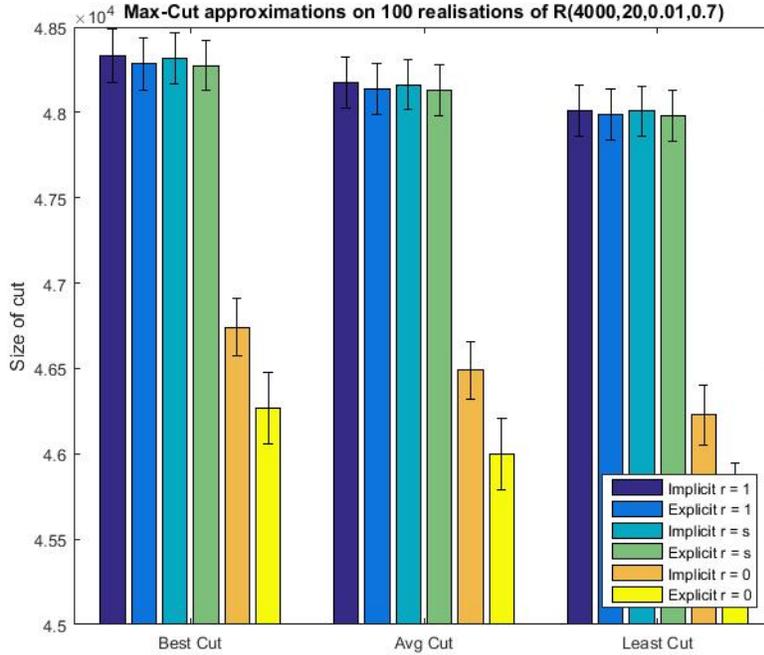
Figure 16. Bar chart of Max-Cut approximations on 100 realisations of $R(4000, 20, 0.01, 0.7)$.

scheme using a spectral truncation method and an Euler method. We also derived a Lyapunov functional for our signless MBO scheme and proved that a rescaled version of this functional Γ-converges to the signless total variation, minimizers of which correspond to maximum cuts.

We tested the resulting (MBO+) algorithm on various graphs using both these signless diffusion solvers, and compared the results and run times with those obtained using the (GW) algorithm. In our tests on realizations of random Erdös-Rényi graphs and on realizations of random modular graphs our (MBO+) algorithm using the spectral method outperforms (GW) with reduced run times. On our examples of scale free graphs (GW) usually gives the best maximum cut approximations, but requires run times that are two orders of magnitude longer than those of (MBO+), which obtains cut sizes within about 2% of those obtained by (GW). Similar conclusions follow from our tests on weighted graphs, that used randomly generated Erdös-Rényi graphs and modular graphs, and some scale free graphs, all with random edge weights. We have also shown that our algorithm using the Euler method can be used on large sparse datasets, with reasonable computation times.

In our tests (and for our parameter choices) we see that (MBO+) using both Δ_1^+ and Δ_s^+ produces larger Max-Cut approximations than Δ_0^+ for all of the graphs that we tested on.

There are still many open questions related to the (MBO+) algorithm, for example questions related to a priori parameter choices (such as τ and K), and performance guarantees. These can be the subject of future research.

REFERENCES

[1] University of Ioannina, Department of Computer Science and Engineering Teaching Resources, http://www.cs.uoi.gr/tsap/teaching/InformationNetworks/data-code.html., Accessed: 2017-03-13.

[2] MIT Strategic Engineering Research Group: Matlab Tools for Network Analysis (2006-2011), http://strategic.mit.edu/docs/matlab_networks/random_modular_graph.m., Accessed: 2017-07-20.

[3] SNAP Datasets: Stanford Large Network Dataset Collection, http://snap.stanford.edu/data, Accessed: 2017-07-01.

[4] R. Albert, H. Jeong and A.-L. Barabási, Internet: Diameter of the world-wide web, *Nature*, (1999), 130–131.

[5] S. M. Allen and J. W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, *Acta Metallurgica*, **27** (1979), 1085–1095.

[6] L. Ambrosio, N. Gigli and G. Savaré, *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*, Springer Science & Business Media, 2008.

[7] A.-L. Barabási, Scale-free networks: A decade and beyond, *Science*, (2002), 412–413.

[8] A.-L. Barabási and E. Bonabeau, Scale-free, *Scientific American*, (2003), 50–59.

[9] F. Barahona, M. Grötschel, M. Jünger and G. Reinhelt, An application of combinatorial optimization to statistical physics and circuit layout design, *Scientific American*, (2003), 50–59.

[10] G. Barles and C. Georgelin, A simple proof of convergence for an approximation scheme for computing motions by mean curvature, *SIAM Journal on Numerical Analysis*, **32** (1995), 484–500.

[11] A. L. Bertozzi and A. Flenner, Diffuse interface models on graphs for classification of high dimensional data, *Multiscale Modeling & Simulation*, **10** (2012), 1090–1118.

[12] B. Bollobás, *Modern Graph Theory*, Graduate Texts in Mathematics, 184. Springer-Verlag, New York, 1998.

[13] A. Braides, *Gamma-convergence for Beginners*, Clarendon Press, 2002.

[14] L. Bronsard and R. V. Kohn, Motion by mean curvature as the singular limit of Ginzburg–Landau dynamics, *Journal of Differential Equations*, **90** (1991), 211–237.

[15] S. Bylka, A. Idzik and Z. Tuza, Maximum cuts: Improvements and local algorithmic analogues of the Edwards–Erdös inequality, *Discrete Mathematics*, **194** (1999), 39–58.

[16] L. Calatroni, Y. van Gennip, C.-B. Schönlieb, H. M. Rowland and A. Flenner, Graph clustering, variational image segmentation methods and Hough transform scale detection for object measurement in images, *Journal of Mathematical Imaging and Vision*, **57** (2017), 269–291.

[17] D. Calvetti, L. Reichel and D. C. Sorensen, An implicitly restarted Lanczos method for large symmetric eigenvalue problems, *Electronic Transactions on Numerical Analysis*, **2** (1994), 1–21.

[18] F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Society, 1997.

[19] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, CUP Archive, 1965.

[20] G. Dal Maso, *An Introduction to Γ-convergence*, in *Progress in Nonlinear Differential Equations and Their Applications*, Birkhäuser, 1993.

[21] M. Desai and V. Rao, A characterization of the smallest eigenvalue of a graph, *Journal of Graph Theory*, **18** (1994), 181–194.

[22] S. Esedoğlu and F. Otto, Threshold dynamics for networks with arbitrary surface tensions, *Communications on Pure and Applied Mathematics*, **68** (2015), 808–864.

[23] J. G. F. Francis, The QR transformation a unitary analogue to the LR transformation—Part 1, *The Computer Journal*, **4** (1961), 265–271.

[24] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, San Francisco, LA: Freeman, 1979.

[25] M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *Journal of the ACM (JACM)*, **42** (1995), 1115–1145.

[26] G. H. Golub and C. F. van Loan, *Matrix Computations*, Fourth edition. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 2013.

[27] M. Grötschel and W. R. Pulleyblank, Weakly bipartite graphs and the Max-Cut problem, *Operations Research Letters*, **1** (1981), 23–27.

[28] J.-M. Guo, A new upper bound for the Laplacian spectral radius of graphs, *Linear Algebra and Its Applications*, **400** (2005), 61–66.

[29] V. Guruswami, Maximum cut on line and total graphs, *Discrete Applied Mathematics*, **92** (1999), 217–221.

[30] F. Hadlock, Finding a maximum cut of a planar graph in polynomial time, *SIAM Journal on Computing*, **4** (1975), 221–225.

[31] Y. Haribara, S. Utsunomiya and Y. Yamamoto, A coherent Ising machine for MAX-CUT problems: Performance evaluation against semidefinite programming and simulated annealing, in *Principles and Methods of Quantum Information Technologies* Springer, **911** (2016), 251–262.

[32] M. Hein, J.-Y. Audibert and U. von Luxburg, Graph Laplacians and their convergence on random neighborhood graphs, *Journal of Machine Learning Research*, **8** (2007), 1325–1368.

[33] H. Hu, T. Laurent, M. A. Porter and A. L. Bertozzi, A method based on total variation for network modularity optimization using the MBO scheme, *SIAM Journal on Applied Mathematics*, **73** (2013), 2224–2246.

[34] C.-Y. Kao and Y. van Gennip, Private communication.

[35] R. M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations*, Springer, (1972), 85–103.

[36] S. Khot, On the power of unique 2-prover 1-round games, in *Proceedings of the thirty-fourth Annual ACM Symposium on Theory of Computing*, ACM, (2002), 767–775.

[37] S. Khot, G. Kindler, E. Mossel and R. O'Donnell, Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?, *SIAM Journal on Computing*, **37** (2007), 319–357.

[38] J. B. Lasserre, A MAX-CUT formulation of 0/1 programs, *Operations Research Letters*, **44** (2016), 158–164.

[39] R. B. Lehoucq and D. C. Sorensen, Deflation techniques for an implicitly restarted Arnoldi iteration, *SIAM Journal on Matrix Analysis and Applications*, **17** (1996), 789–821.

[40] E. Merkurjev, T. Kostic and A. L. Bertozzi, An MBO scheme on graphs for classification and image processing, *SIAM Journal on Imaging Sciences*, **6** (2013), 1903–1930.

[41] B. Merriman, J. Bence and S. Osher, *Diffusion Generated Motion by Mean Curvature*, UCLA Department of Mathematics CAM report CAM 06–32, 1992.

[42] B. Merriman, J. Bence and S. Osher, Diffusion generated motion by mean curvature, *AMS Selected Letters, Crystal Grower's Workshop*, (1993), 73–83.

[43] H. D. Mittelmann, The state-of-the-art in conic optimization software, in *Handbook on Semidefinite, Conic and Polynomial Optimization* Springer, **166** (2012), 671–686.

[44] L. Modica and S. Mortola, Un esempio di Γ^--convergenza, *Boll. Un. Mat. Ital. B (5)*, **14** (1977), 285–299.

[45] L. Modica, The gradient theory of phase transitions and the minimal interface criterion, *Archive for Rational Mechanics and Analysis*, **98** (1987), 123–142.

[46] B. Mohar, The Laplacian spectrum of graphs, in *Graph Theory, Combinatorics, and Applications, Volume 2*, Wiley, (1991), 871–898.

[47] R. J. Radke, *A Matlab Implementation of the Implicitly Restarted Arnoldi Method for Solving Large-scale Eigenvalue Problems*, Masters thesis, Rice University, 1996.

[48] W. Rudin, *Principles of Mathematical Analysis*, McGraw-hill New York, 1964.

[49] J.-L. Shu, Y. Hong and K. Wen-Ren, A sharp upper bound on the largest eigenvalue of the Laplacian matrix of a graph, *Linear Algebra and Its Applications*, **347** (2002), 123–129.

[50] D. C. Sorensen, Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations, in *Parallel Numerical Algorithms*, Springer, **4** (1997), 119–165.

[51] J. A. Soto, Improved analysis of a Max-Cut algorithm based on spectral partitioning, *SIAM Journal on Discrete Mathematics*, **29** (2015), 259–268.

[52] L. Trevisan, G. B. Sorkin, M. Sudan and D. P. Williamson, Gadgets, approximation, and linear programming, *SIAM Journal on Computing*, **29** (2000), 2074–2097.

[53] L. Trevisan, Max-cut and the smallest eigenvalue, *SIAM Journal on Computing*, **41** (2012), 1769–1786.

[54] R. H. Tütüncü, K.-C. Toh and M. Todd, Solving semidefinite-quadratic-linear programs using SDPT3, *Mathematical Programming*, **95** (2003), 189–217.

[55] Y. van Gennip and A. L. Bertozzi, Γ-convergence of graph Ginzburg-Landau functionals, *Advances in Differential Equations*, **17** (2012), 1115–1180.

[56] Y. van Gennip, N. Guillen, B. Osting and A. L. Bertozzi, Mean curvature, threshold dynamics, and phase field theory on finite graphs, *Milan Journal of Mathematics*, **82** (2014), 3–65.

[57] Y. van Gennip, An MBO scheme for minimizing the graph Ohta–Kawasaki functional, Journal Of Nonlinear Science, 2018.

[58] U. von Luxburg, A tutorial on spectral clustering, *Statistics and computing*, **17** (2007), 395–416.

[59] X.-D. Zhang, The signless Laplacian spectral radius of graphs with given degree sequences, *Discrete Applied Mathematics*, **157** (2009), 2928–2937.

E-mail address: Blaine.Keetch@nottingham.ac.uk
E-mail address: Y.VanGennip@nottingham.ac.uk